

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky**

**Simulační hra pro výuku podnikání
Simulation game for business education**

2016

Bc. Marek Berger

Zadání diplomové práce

Student:

Bc. Marek Berger

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Simulační hra pro výuku podnikání
Simulation Game for Business Education

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem této diplomové práce je sestavit webovou aplikaci pro simulační hru pro výuku podnikání. Hra bude zaměřena převážně na lokální komerční prostředí a bude zohledňovat zvyklosti a požadavky trhu. Cílem je sestavit samostatně fungující expertní systém, který bude simulovat nabídku a poptávku, změny trhu, výkyvy a anomálie, problémové situace v podnikání. Součástí hry bude i vzdělávací obsah, který uživateli pomůže s osvojením terminologie, postupů a procesů souvisejících s podnikáním, logistikou, výrobou, lidskými zdroji, managementem a marketingem.

1. Seznamte se s problematikou simulačních her. Proved'te analýzu existujících řešení, navrhněte vhodné hodnotící a porovnávací kritéria a sestavte přehled funkcí, možností a principů (hry) vybraných konkurenčních řešení.
2. Na základě získaných poznatků definujte cíle vlastního řešení, popište model simulace chování trhu, aktivity uživatelů, model hodnocení a způsob publikace vzdělávacího obsahu.
3. Na základě získaných poznatků proved'te analýzu simulační hry, kterou budete implementovat.
4. Proved'te ve vhodně zvoleném prostředí implementaci webové aplikace, s možnou návazností na mobilní zařízení. Exekutivní část implementujte jako aplikaci optimalizovanou na výkon na straně serveru.
5. Při návrhu uživatelského rozhraní se zaměřte na intuitivnost ovládání, dostatečnou nápovědu v podobě textu nebo instruktážních videí, tak aby uživatel samostatně bez pomoci zvládl všechny kroky a postupy v systému implementované.
6. Proved'te pilotní běh hry. Na základě získaných poznatků optimalizujte výsledné řešení.
7. Dosažené výsledky prezentujte v rámci hodnocených konferencí nebo časopisů zabývajících se problematikou simulací, elektronického vzdělávání a elektronické komerce. Srovnajte výsledné řešení s existujícími systémy simulačních her.

Seznam doporučené odborné literatury:

Analýza konkurenčních projektů:

<http://www.markethero.org>

<http://www.marketplace-live.com/simulations/strategic-e-commerce.php>

<http://epiphanylearning.co.in/masquerade-the-e-commerce-business-simulation-game/>

<http://simventure.co.uk/about-us/clients>

Dále dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

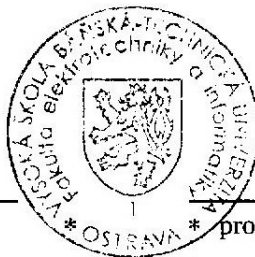
Vedoucí diplomové práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *25. dubna 2016*



.....
podpis studenta

Poděkování

Rád bych zde poděkoval Ing. Radoslavu Fasugovi Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Abstrakt

Tématem této diplomové práce je problematika simulačních her zaměřující se na oblast podnikání. Cílem práce je navrhnout, implementovat a otestovat vlastní řešení, jež bude simulovat chování trhu, procesy firemní infrastruktury a důsledky strategických rozhodnutí podnikatele a to v přívětivém grafickém rozhraní a se začleněním výukových i herních faktorů. První část práce je věnována teoretickému základu pro vlastní implementaci. Jsou rozebrány principy podnikání ve zjednodušeném rozsahu s ohledem na existenci možných způsobů jejich implementace, a také principy a doporučené postupy pro tvorbu her. Dostatečný prostor je věnován i popisu stávajících řešení. Druhá část je zaměřena na praktickou stránku práce. Kapitoly jsou strukturovány dle etap vývoje od prvotní vize přes analýzu a návrh až k implementaci a testování.

Klíčová slova

Simulace podnikání, webová strategická hra, výuka podnikání, responzivní design

Abstract

The theme of this thesis is the issue of simulation games focusing on business. The aim is to design, implement and test a custom solution that will simulate the behavior of market processes, enterprise infrastructure and consequences of strategic decisions of businessmen in friendly graphical interface and the inclusion of educational and game factors. The first part is devoted to theoretical basis for their implementation. Business principles are described in a simplified range with respect to the existence of possible ways of their implementation, as well as the principles and best practices for creating games. Ample space is devoted to a description of existing solutions. The second part focuses on the practical aspects of the thesis. The chapters are structured according to the stages of development from initial vision through analysis and design to implementation and testing.

Key words

Business simulation, web strategy game, business learning, responsive design

Obsah

Seznam použitých zkratk	- 15 -
Seznam ilustrací, tabulek a rovnic	- 16 -
Úvod	- 19 -
1 Podnikání v České republice	- 21 -
1.1 Předpoklady úspěchu začínající firmy	- 21 -
1.2 Rozvaha	- 21 -
1.3 Právní formy podnikání	- 22 -
1.3.1 Podnikání fyzických osob	- 22 -
1.3.2 Podnikání právnických osob	- 23 -
1.4 Marketing	- 24 -
1.4.1 Poznávací stránka marketingu	- 24 -
1.4.2 Realizační stránka marketingu	- 24 -
1.4.3 Cenová politika	- 24 -
1.5 Lidské zdroje	- 25 -
1.5.1 Mzda	- 25 -
2 Teorie herního designu	- 27 -
2.1 Co je to herní design?	- 27 -
2.2 Schopnosti herního designéra	- 27 -
2.3 Herní zážitek	- 28 -
2.4 Herní elementy	- 28 -
2.5 Tematika	- 29 -
2.6 Mentální schopnosti hráče	- 29 -
2.6.1 Modelování	- 29 -
2.6.2 Soustředění	- 29 -
2.6.3 Empatie	- 30 -
2.6.4 Představivost	- 30 -
2.7 Herní design strategické hry	- 30 -
2.7.1 Zásady při tvorbě designu	- 30 -
3 Stávající systémy pro výuku podnikání	- 31 -
3.1 Market Hero	- 31 -
3.2 Marketplace Live	- 32 -

3.3	SimVenture.....	- 34 -
3.4	Travian	- 36 -
4	Vize vlastního řešení	- 39 -
4.1	Postava podnikatele.....	- 39 -
4.2	Mapa prostředí.....	- 39 -
4.3	Mapa regionu.....	- 40 -
4.3.1	Budovy.....	- 40 -
4.4	Mapa nemovitosti	- 41 -
4.4.1	Vybavení nemovitosti	- 41 -
4.4.2	Standardní vybavení v budově.....	- 41 -
4.4.3	Nákup a prodej vybavení	- 42 -
4.5	Živnostenské oprávnění.....	- 42 -
4.6	Trh	- 42 -
4.6.1	Kapacita trhu.....	- 42 -
4.6.2	Geografická segmentace trhu.....	- 42 -
4.6.3	Demografická a psychografická segmentace trhu	- 42 -
4.6.4	Průzkum trhu a průzkum konkurence	- 43 -
4.7	Produkty	- 43 -
4.7.1	Parametry produktu	- 44 -
4.7.2	Životní cyklus produktu ve společnosti	- 44 -
4.8	Simulace prodeje	- 44 -
4.8.1	Návštěvnost veřejné budovy	- 45 -
4.8.2	Trendy.....	- 45 -
4.8.3	Cena produktu.....	- 46 -
4.8.4	Poptávka zákazníka	- 47 -
4.8.5	Hodnocení zákazníka	- 47 -
4.9	Zaměstnanci.....	- 47 -
4.9.1	Schopnosti zaměstnance	- 47 -
4.9.2	Najímání zaměstnanců.....	- 49 -
5	Analýza požadavků.....	- 51 -
5.1	Struktura systému	- 51 -
5.2	Aktéři systému.....	- 51 -
5.3	Případy užití	- 51 -

5.3.1	Výchozí část	- 52 -
5.3.2	Herní uživatelská část	- 53 -
5.3.3	Herní systémová část	- 60 -
5.3.4	Administrační část	- 61 -
6	Návrh aplikace	- 63 -
6.1	Doménový model	- 63 -
6.1.1	Výchozí část	- 63 -
6.1.2	Herní sdílená část	- 64 -
6.1.3	Exteriér	- 65 -
6.1.4	Interiér	- 67 -
6.1.5	Společnost	- 69 -
6.1.6	Zboží	- 71 -
6.1.7	Lidé	- 74 -
6.1.8	Trh	- 75 -
6.2	Procedurální náhled	- 77 -
6.2.1	Generování objednávek	- 77 -
6.2.2	Generování úsilí	- 78 -
6.2.3	Generování úsilí na markování objednávek	- 79 -
6.2.4	Snižování a zvyšování množství zásob	- 80 -
6.2.5	Aktualizace pracovních poměrů	- 81 -
6.3	Serverová část aplikace	- 82 -
6.3.1	Přístup k datům	- 82 -
6.3.2	Služby	- 85 -
6.4	Klientská část aplikace	- 87 -
6.5	Grafické rozhraní	- 89 -
6.5.1	Izometrická mapa	- 90 -
6.5.2	Logo	- 92 -
7	Implementace	- 93 -
7.1	Použité technologie	- 93 -
7.1.1	ASP.NET MVC 5	- 93 -
7.1.2	Microsoft SQL Server	- 93 -
7.1.3	Entity framework	- 93 -
7.1.4	KnockoutJS	- 93 -

7.1.5	EaselJS.....	- 94 -
7.1.6	Bootstrap.....	- 94 -
7.1.7	Blender.....	- 94 -
7.2	Komponenty	- 94 -
7.3	Struktura webové aplikace	- 95 -
7.4	Diagram nasazení	- 95 -
8	Výsledky a testování.....	- 97 -
8.1	Možnosti budoucího rozšíření	- 99 -
8.1.1	Rozšíření sortimentu	- 99 -
8.1.2	Alternativní motivy.....	- 99 -
8.1.3	Mobilní aplikace	- 100 -
8.1.4	Placené služby	- 100 -
8.2	Testování	- 100 -
8.3	Propagace	- 100 -
	Závěr.....	- 101 -
	Použitá literatura.....	- 103 -
	Seznam příloh.....	- 105 -

Seznam použitých zkratk

Zkratka	Význam
ASP.NET	Active server pages .NET
CRUD	Create read update delete
CSS	Cascading Style Sheets
DDL	Data definition language
DI	Dependency injection
DML	Data manipulation language
GOF	Gang of four
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
JSON	Javascript object notation
MVC	Model view controller
MVVM	Model view viewmodel
OSVČ	Osoba samostatně výdělečně činná
PR	Public relations
S. R. O.	Společnost s ručením omezeným
SQL	Structured query language
T-SQL	Transact - SQL

Seznam ilustrací, tabulek a rovnic

Obrázek 3.1 – Založení firmy v aplikaci Market Hero	- 31 -
Obrázek 3.2 – Řízení firmy v aplikaci Market Hero	- 32 -
Obrázek 3.3 - Ukázka aplikace Marketplace Live	- 33 -
Obrázek 3.4 – Navigační menu v aplikaci SimVenture	- 34 -
Obrázek 3.5 – Odehrání tahu v aplikaci SimVenture	- 35 -
Obrázek 3.6 – Návrh produktu v aplikaci SimVenture	- 36 -
Obrázek 3.7 – Mapa vesnic ve hře Travian	- 37 -
Obrázek 3.8 – Centrum vesnice ve hře Travian	- 38 -
Obrázek 5.1- Diagram případu užití výchozí části	- 52 -
Obrázek 5.2 – Diagram případu užití herní uživatelské části	- 54 -
Obrázek 5.3 - Diagram případu užití herní systémové části	- 61 -
Obrázek 5.4 - Diagram případu užití administrační části	- 62 -
Obrázek 6.1 – Doménový model – Herní sdílená část	- 64 -
Obrázek 6.2 – Doménový model – exteriér	- 66 -
Obrázek 6.3 – Doménový model – interiér	- 68 -
Obrázek 6.4 – Doménový model – společnost	- 70 -
Obrázek 6.5 – Doménový model – zboží	- 72 -
Obrázek 6.6 – Doménový model – lidé	- 74 -
Obrázek 6.7 – Doménový model – trh	- 76 -
Obrázek 6.8 – Diagram aktivit pro generování objednávek	- 78 -
Obrázek 6.9 – Diagram aktivit pro generování úsilí	- 79 -
Obrázek 6.10 – Diagram aktivit pro generování úsilí na markování objednávek	- 80 -
Obrázek 6.11 – Diagram aktivit pro snižování a zvyšování množství zásob	- 80 -
Obrázek 6.12 – Diagram aktivit pro aktualizaci pracovních poměrů	- 81 -
Obrázek 6.13 – Architektura MVC	- 82 -
Obrázek 6.14 – Třídní diagram struktury repositářů	- 84 -
Obrázek 6.15 – Třídní diagram služeb	- 86 -
Obrázek 6.16 – Architektura MVVM	- 87 -
Obrázek 6.17 – Třídní diagram struktury viewmodelů	- 88 -
Obrázek 6.18 – Grafické rozhraní – výchozí část	- 89 -
Obrázek 6.19 – Grafické rozhraní – herní uživatelská část	- 90 -
Obrázek 6.20 – Izometrická mapa	- 91 -
Obrázek 6.21 – Logo	- 92 -
Obrázek 7.1 – Diagram komponent	- 95 -
Obrázek 7.2 – Diagram nasazení	- 96 -
Obrázek 8.1 – Výběr hry	- 97 -
Obrázek 8.2 – Mapa regionu	- 97 -
Obrázek 8.3 – Mapa nemovitosti	- 98 -
Obrázek 8.4 – Okno zaměstnance	- 98 -
Obrázek 8.5 – Okno výběru produktu	- 99 -

<i>Tabulka 1.1 – Struktura rozvahy.....</i>	<i>- 22 -</i>
<i>Tabulka 4.1 – Ukázka matice trendů.....</i>	<i>- 43 -</i>
<i>Tabulka 4.2 – Aktivita</i>	<i>- 48 -</i>
<i>Rovnice 4.1 – Výpočet prodejnosti produktu.....</i>	<i>- 45 -</i>
<i>Rovnice 4.2 – Výpočet faktoru trendů</i>	<i>- 46 -</i>
<i>Rovnice 4.3 – Výpočet faktoru ceny produktu.....</i>	<i>- 46 -</i>
<i>Rovnice 4.4 – Výpočet množství zkušeností pro další úroveň zaměstnance</i>	<i>- 48 -</i>
<i>Rovnice 6.1 – Převod izometrických souřadnic</i>	<i>- 91 -</i>

Úvod

Jistě podstatná část obyvatel měla v jistém životním období sen o tom, že se stanou podnikateli a budou vézt vlastní společnost. Motivujících faktorů je v tomto směru mnoho, od naivních dětských přání, až po touhu být finančně nezávislý, případně touhu naplnit své vize. Většina těchto snů ovšem zůstane i nadále jen snem. Dokazují to statistická čísla, která naznačují, že se situace zhoršuje, a že čistého přírůstku začínajících podnikatelů je stále méně. Lidé mají své bariéry, které jim brání své sny zrealizovat. Bojí se proto, že nemají k realizaci dostatečný kapitál nebo postrádají dobrý podnikatelský nápad, který by obstál na trhu s konkurencí, ale také i proto, že nejsou dostatečně informováni o principech a zákonných aspektech v podnikání.

S nástupem informačních technologií se otevřely nové možnosti poskytování a ztvárnění informací. Začaly vznikat počítačové hry, které uživatelům umožňují uniknout z reálného světa a pomocí vysoké míry interakce, kterou jiná média neumožňují, se vžít do role někoho jiného. Počítačové hry jsou ovšem z úhlu pohledu rozvoje vlastní osobnosti vnímány negativně a u určitých typů her i oprávněně. Ovšem existují i hry, které po uživateli vyžadují logické, taktické případně kreativní myšlení a takové již mohou jistým způsobem k osobnímu rozvoji přispívat. S technologickými možnostmi dnešní doby tedy nic nebrání tomu, aby mohl uživatel uniknout z reálného světa a vrátit se do jeho simulované podoby, ovšem v roli podnikatele.

Právě návrh a implementace prostředí simulujícího fiktivní marketingový trh, fungování firemní infrastruktury a důsledky strategických rozhodnutí podnikatele, v jehož roli bude uživatel vystupovat, je předmětem této diplomové práce. Plánovaná simulační hra bude značně inspirována reálným prostředím, ale jelikož je prakticky nemožné ztvárnit některé aspekty, například vnímání a rozhodování zákazníků při nákupu produktu, bude v těchto ohledech značně zjednodušena. Měla by být ovšem do jisté míry zachována jednak simulační stránka a jednak výuková stránka s návazností na reálné podnikání v České republice. Jelikož jde v podstatě o počítačovou simulační hru, je nutné brát v úvahu i aspekt zábavnosti a soutěživosti. Uživatel, který chce získat informace o podnikání, si jistě představí jiný způsob jejich prezentace a zamíří raději k psanému slovu, kde se může zaměřit přímo na to, co ho zajímá. Primárním cílem tedy není vytvořit simulaci, která uživatele naučí podnikat, nýbrž vytvořit simulaci, která bude pro uživatele zábavná i poučná a jeho mysl bude motivována i vyzývána k opětovnému užívání.

Práce se nevěnuje oblasti podnikání do detailů, začleněny jsou pouze aspekty, které lze názorným způsobem simulovat a jejichž ztvárnění není pro uživatele matoucí a frustrující. Shrnutí začleněné oblasti podnikání je předmětem první kapitoly. Ve druhé kapitole jsou popsány doporučené praktiky při tvorbě počítačových her a to nikoli z technického nýbrž z psychologického pohledu. Existující strategické simulační hry a aplikace jsou popsány ve třetí kapitole, jsou charakterizovány jejich klady a zápory a nastíněna funkcionalita jejich simulačních modelů. Od čtvrté kapitoly se práce zaměřuje na realizaci vlastního řešení počínaje popisem vlastní vize, kde je určeno jaké funkční bloky mají být obsaženy a jak by tyto bloky měly působit na uživatele. Obsahem páté kapitoly je převod myšlenek vize do více systematického prostředí pomocí jejich analýzy. Následný návrh a implementace vlastního řešení jsou probírány v šesté a sedmé kapitole. Poslední osmá kapitola pak prezentuje dosažené výsledky, popisuje průběh testování a zdůrazňuje body hodné dalšího rozšíření.

1 Podnikání v České republice

Pojem podnikání je v obchodním zákoníku i živnostenském zákoně definován jako „...soustavná činnost prováděná samostatně podnikatelem vlastním jménem a na vlastní odpovědnost za účelem dosažení zisku“. Podnikatelem se člověk nenarodí, nýbrž může se jím stát každý. Ovšem vše záleží jen na jeho vůli a zapálení pro věc, což jsou vlastnosti, které se nedají naučit. Co se ale naučit dá, jsou procesy probíhající v podnikání. Úvodní kapitola shrnuje tyto procesy a vytváří teoretický základ pro jejich implementaci do virtuálního prostředí [1].

1.1 Předpoklady úspěchu začínající firmy

Před zahájením podnikatelské činnosti, by si měl podnikatel ujasnit své cíle a vyhodnotit možnosti jak jich dosáhnout. Podnikatel je osoba jako každá jiná, má tedy své silné i slabé stránky. Některé osobní předpoklady pro podnikání může mít tato osoba vrozené, jiné se musí naučit, a to jak studiem, tak ziskem praktických zkušeností. Již před zahájením podnikání by si ovšem podnikatel měl uvědomovat své osobní předpoklady a měl by být schopen odpovědět na následující otázky:

- Jaké jsou mé osobní a podnikatelské (jak krátkodobé tak dlouhodobé) cíle, a do jaké míry se shodují?
- Mám již zkušenosti s podnikáním, a jakých výsledků jsem zatím dosáhl?
- Jsem schopen realizovat své cíle a vézt k nim i své zaměstnance?
- Dokážu čelit eventuální ztrátě případně jiným nepříjemným situacím a převzít zodpovědnost za svá rozhodnutí?
- Mám dostatečné odborné znalosti, nebo si potřebuji doplnit vzdělání?

Dostatečné odhodlání pro podnikání mimo jiné utvrzuje dobrý podnikatelský nápad. Předpokladem úspěchu je jeho reálná životaschopnost a jeho možné uplatnění na cílovém trhu. Posuzovat kvalitu nápadu je možné posouzením odpovědí například na tyto otázky:

- Jaký je charakter výrobku nebo služby a jaké potřeby zákazníků uspokojuje?
- Kdo jsou konkurenti, jaké produkty nabízí a za jakou cenu, a proč je můj produkt lepší než jejich?
- Jak velký a perspektivní je cílový trh a o kolik produktů bude v určitém období zájem?
- Co nás vedlo k myšlence, že náš produkt budou zákazníci chtít a budou ho kupovat?

Pro lepší odhalení slabin a identifikaci silných stránek podnikatelského nápadu je vhodné sestavit podnikatelský plán, kde bude vymezen okruh výrobků nebo služeb, jež firma nabízí, identifikování potenciální zákazníci na cílovém trhu, zmapována konkurence a v neposlední řadě sepsány finanční možnosti [1].

1.2 Rozvaha

Rozvaha představuje základní účetní výkaz a slouží k evidenci majetku a zdrojů použitých k jeho financování. Z hlediska účetnictví jsou tyto položky označovány jako aktiva a pasiva. První rozvahu by měl podnikatel zpracovat už při založení firmy pro vytvoření lepší představy o finanční zátěži.

Majetek firmy se dle doby setrvání v jedné majetkové formě člení na dlouhodobý a oběžný majetek. Dlouhodobý majetek je jen částečně opotřebováván a je používán po několik let. Řadí se k němu například nemovitosti, výrobní stroje, licence k softwaru nebo cenné papíry. Do oběžného majetku spadají zásoby zboží nebo materiálu, pohledávky, peníze na účtech a podobně.

Pořízení majetku může být financováno z vlastních zdrojů tedy z vlastního kapitálu, nebo pomocí cizích zdrojů, ke kterým se řadí například bankovní úvěry [1].

Příklad struktury rozvahy uvádí Tabulka 1.1.

Aktiva	Pasiva
Dlouhodobý majetek	Vlastní kapitál
Nehmotný dlouhodobý majetek	Základní kapitál
Hmotný dlouhodobý majetek	
Finanční dlouhodobý majetek	Cizí zdroje
Oběžný majetek	Závazky
Zásoby	Bankovní úvěry a výpomoci
Pohledávky	Půjčky
Krátkodobý finanční majetek	
Celkem	Celkem

Tabulka 1.1 – Struktura rozvahy

1.3 Právní formy podnikání

Volba právní formy podnikání je dlouhodobě působící rozhodnutí. Podnikatel by tedy měl mít z dlouhodobého hlediska jasno v tom, co chce dělat a bude se rozhodovat na základě následujících otázek:

- Jaký je způsob a rozsah ručení?
- Kdo má oprávnění k řízení?
- Jaké jsou nároky na počáteční kapitál?
- Jaká je administrativní náročnost založení a vedení firmy?
- Jaké jsou možnosti získávání finančních prostředků?

Obchodní zákoník připouští podnikání fyzických a podnikání právnických osob.

1.3.1 Podnikání fyzických osob

Fyzická osoba je v kontextu podnikání v České republice označována jako osoba samostatně výdělečně činná (OSVČ). Tato osoba musí před zahájením podnikání získat živnostenské oprávnění. To může být podmíněno odbornou způsobilostí v závislosti na typu živnosti, kterou chce podnikatel

provozovat. Dle způsobu vzniku se živnosti dělí na ohlašovací a koncesované. Ohlašovací vznikají na základě ohlášení a jsou osvědčeny výpisem ze živnostenského rejstříku. Ty se dále dělí na:

- Řemeslné živnosti: Pro získání oprávnění je nutné se prokázat výučním listem, maturitou nebo diplomem v oboru, případně šestiletou praxí. Příkladem řemeslné živnosti je řeznictví, zednictví nebo malířství.
- Vázané živnosti: Pro získání oprávnění je nutné prokázat odbornou způsobilost, která je stanovena přílohou živnostenského zákona. Příkladem jsou masérské služby, provozování autoškoly a podobně.
- Volné živnosti: Pro získání oprávnění nepotřebuje podnikatel žádnou odbornou způsobilost. Rozsah své působnosti si vybere sám ze seznamu osmdesáti činností stanovených přílohou živnostenského zákona. Příkladem je provozování maloobchodu nebo velkoobchodu, ubytovacích služeb a podobně.

Koncesované živnosti vznikají, kromě prokázání odborné způsobilosti, na základě kladného vyjádření příslušného orgánu státní správy. Příkladem je provozování pohřební služby nebo cestovní kanceláře.

Forma podnikání fyzických osob je vhodná pro začínající podnikatele, nebo pro ty, kteří v podnikání nevidí hlavní zdroj příjmů. Po splnění registračních úkonů je možné zahájit podnikání téměř ihned a bez základního kapitálu. Administrativní náročnost na vedení podniku je nízká a do jisté výše obratu postačí jen vedení daňové evidence. Nevýhodou je, že podnikatel za svou podnikatelskou činnost ručí celým svým majetkem [1].

Založit živnost je možné na kterémkoliv živnostenském úřadě vyplněním jednotného registračního formuláře, který kromě registrace živnosti umožňuje i registraci na daň z příjmu fyzických osob, která aktuálně činí 15%, na sociální a zdravotní pojištění a na další volitelné záležitosti například na plátce DPH. Do formuláře je nutné mimo jiné uvést zvolené činnosti podnikání a místo vykonávání těchto činností. Stejný formulář je používán i pro aktualizaci těchto údajů. Při první registraci je ještě nutné uhradit správní poplatek ve výši tisíc korun [2].

1.3.2 Podnikání právnických osob

Pokud se podnikatel rozhodne podnikat jako právnická osoba, musí počítat s náročnější administrativou při zahájení i vedení společnosti. Všechny typy právnických osob musí být zapsány v obchodním rejstříku, který vymezuje tento typ na osobní společnost (veřejná obchodní společnost, komanditní společnost), kapitálovou společnost (společnost s ručením omezeným, akciová společnost) a družstvo.

Nejčastějším typem právnické osoby v České republice je společnost s ručením omezeným (s. r. o.). Společníky společnosti mohou být jak právnické tak fyzické osoby. Všichni společníci ručí za závazky společnosti společně a nerozdílně do výše souhrnu nesplacených částí vkladů. Společnost musí vézt plnohodnotné účetnictví [1].

S. r. o. je kapitálová společnost a pro její založení je nutné složit základní kapitál. Jeho minimální výše je od roku 2014 pouhá jedna koruna. Společnost je definována pomocí společenské smlouvy, ve které je určeno sídlo firmy, určení společníci a jednatele, vymezen předmět podnikání a v neposlední řadě určena výše základního kapitálu a výše vkladu každého společníka. Uzavřená

společenská smlouva formou notářského zápisu, se spolu s potřebnými živnostenskými listy, osvědčeními o oprávněném užívání provozoven firmy a zaplacenými základními vklady, předkládají rejstříkovému soudu jako návrh na zápis do obchodního rejstříku. Po zapsání do obchodního rejstříku je společnost založena. Po založení je povinností provést registraci k dani z příjmů právnických osob, jež je aktuálně zákonem stanovena na 19%, a pokud má firma nějaké zaměstnance, i k platbě sociálního a zdravotního pojištění [3].

1.4 Marketing

Pojem marketing je často mylně spojován jen s reklamními a propagačními aktivitami společnosti. Podstata marketingu ovšem obsahuje daleko širší spektrum. V odborné literatuře existuje pro tento pojem celá řada definic, ovšem jejich společný jmenovatel je orientace na zákazníka. Podstatou marketingu je získat informace o trhu, na kterém firma působí, a tedy poznat své potenciální zákazníky, a na základě těchto informací co nejvíce ovlivňovat trh v zájmu firemních cílů.

1.4.1 Poznávací stránka marketingu

Prostředí, ve kterém firma působí, se z pohledu marketingu skládá z mikroprostředí a makroprostředí. Mikroprostředí tvoří samotná firma, tedy její funkční oblasti ovlivňující prodejnost produktů nebo služeb, ale také externí zainteresované subjekty tedy zákazníci, konkurenti, dodavatelé a zprostředkovatelé. Makroprostředí naproti tomu představuje téměř nekontrolovatelné vlivy působící na mikroprostředí, a firma je bere jako vymezující hranice jednotlivých segmentů trhu. Ty lze dělit z demografického, společenského, ekonomického, technologického, přírodního a politického hlediska.

Způsobů jak získávat informace o trhu je několik. Nejjednodušší metodou je využití již publikovaných údajů shromážděných pro jiné účely. Pokud chce firma získat konkrétnější informace, musí je ve většině případů nashromáždit sama a to jak pouhým pozorováním, tak experimentováním a dotazováním.

1.4.2 Realizační stránka marketingu

Na základě znalosti trhu, může firma vytvořit svou marketingovou strategii, počínaje identifikací a volbou cílových segmentů zákazníků, přes odhalení konkurenčních výhod pro určení tržní pozice v povědomí zákazníků, až k volbě marketingového mixu, tvořící vzájemně kooperující marketingové nástroje. Nejpoužívanější a nejznámější marketingový mix je označován jako 4P a tvoří ho produkt (product), cena (price), distribuce (place), propagace (promotion).

1.4.3 Cenová politika

Ke stanovení ceny produktu nebo služby jsou uplatňovány tři všeobecné přístupy, a sice přístup založený na firemních nákladech, přístup založený na poptávce a přístup založený na konkurenci. Velká část firem používá první zmíněný přístup, kdy celkové náklady představují spodní hranici pro stanovení ceny. Pro přístup založený na poptávce je nutný dostatečný spotřebitelský průzkum. Cena je totiž stanovena na základě hodnoty produktu nebo služby vnímané zákazníkem. Horní hranice je maximální cena, jakou zákazník ke koupi akceptuje. U posledního přístupu je nutná dobrá znalost konkurentů a jejich strategií. Chce-li podnikatel získat větší podíl na trhu, měl by stanovit cenu pod úrovní konkurence, naopak pokud má firma unikátní produkt nebo pokud podnikatel

očekává přizpůsobení ze strany konkurence, může stanovit cenu nad úroveň. Pokud je trh relativně nasycen, je vhodné stanovit cenu srovnatelnou s konkurencí [1].

1.5 Lidské zdroje

Materiální a finanční zdroje jsou samy o sobě k ničemu, pokud ve firmě není nikdo, kdo by je využíval. S přibývajícím počtem zaměstnanců je pak nezbytné jejich řízení a vedení. V rámci firmy je každý z nich zařazen na určité pracovní místo, které je vytvořeno na základě potřebných úkonů. Zaměstnanec může být na konkrétní pracovní místo vybírán podle vzdělání, praxe, dovedností nebo podle fyzických a duševních dispozic.

Jelikož se společnost a technologie neustále vyvíjí, je nutné zajistit svým pracovníkům dostatek možností k prohlubování svých znalostí a dovedností.

1.5.1 Mzda

Výše mzdy je určována na základě výkonu a zásluhách pracovníka pomocí mzdových forem. Nejběžnější je časová mzdová forma, kdy je pracovník odměňován za počet odpracovaných hodin. Dále může být odměňován za plnění úkolu, za splnění denní kvóty nebo formou provize a podobně.

Mzda sjednaná mezi zaměstnancem a zaměstnavatelem je hrubou mzdou. Z pohledu zaměstnavatele je ovšem vyplacená částka vyšší, jelikož musí za zaměstnance odvézt zdravotní a sociální pojištění ve výši 25% a 9% z hrubé mzdy. Výsledná částka je označována jako superhrubá mzda. Z pohledu zaměstnance je jeho skutečně obdržená částka, tedy čistá mzda, snížena o sociální a zdravotní pojištění ve výši 6,5% a 4,5% z hrubé mzdy, a o daň z příjmu fyzické osoby ve výši 15% ze superhrubé mzdy [4].

2 Teorie herního designu

Jelikož by navrhovaná aplikace měla být jak naučná, tak i zábavná, bude tato kapitola věnována teorii herního designu. Budou zde z psychologického hlediska rozebrány herní principy a mechanismy, které by uživatele, tedy hráče, měly motivovat k dalšímu postupu.

2.1 Co je to herní design?

Herní design je proces rozhodování co by ve hře mělo být obsaženo a jak by to mělo fungovat. Není potřeba žádného zvláštního vybavení. Herní design lze vytvořit jednoduše jen v hlavě. Obvykle je ale třeba tyto myšlenky zaznamenat, aby nebyly zapomenuty a aby bylo možné je prezentovat ostatním členům vývojového týmu.

Herní designéři nemusí být nutně počítačoví programátoři. V první řadě existuje mnoho her, které nevyžadují výpočetní technologie, například karetní hry, deskové hry a podobně. Při vývoji takové hry, nemusí designér vůbec s počítačem pracovat. Obdobně při vývoji počítačové hry nemusí být obeznámen s technickými detaily jejího jádra a může se při práci zaměřit pouze na principy a vizuální koncept.

Je nutné rozeznávat role herního vývojáře a herního designéra. Herní vývojář je osoba, která je jakkoliv zainteresována na vývoji dané hry. Může to být inženýr, animátor, 3D grafik, skladatel, scénárista, producent, ale i designér. Herní designér je tedy jednou specifickou rolí při vývoji hry [5].

2.2 Schopnosti herního designéra

Dobrý herní designér se musí orientovat v mnoha odvětvích. Navíc různé hry mohou po designérovi vyžadovat různé znalosti. Zde je výpis některých nejdůležitějších schopností:

- Animace: Designér nemusí umět animovat, ale musí vědět o možnostech a omezeních této disciplíny.
- Antropologie a psychologie: Designér musí studovat cílovou skupinu uživatelů dané hry, odhadovat jejich touhy a ty potom při návrhu zohlednit. Musí udržet uživatele motivovaného a postarat se mu o dostatečné množství zábavy.
- Architektura: Znalosti architektury designér použije při návrhu herního světa.
- Kinematografie: V některých hrách jsou obsaženy video sekvence, jež zpříjemní herní zážitek. I s tímto konceptem by měl být designér obeznámen.
- Ekonomika: Ve většině her je možné najít nějaký ekonomický systém. I když jde například jen o shromažďování jistých surovin, znalosti ekonomiky se designérovi mohou hodit.
- Historie: Hra může být založena na historických událostech nebo používat historické objekty.
- Matematika: Každá počítačová hra funguje na matematických principech, ať už jde o počítání skóre nebo vykreslování grafiky.
- Hudba a zvuky: Hudba a zvuky navozují tematickou atmosféru.

- Grafický návrh: Představivost je jednou z nejdůležitějších vlastností designéra. V tomto případě jde o vizuální představivost umožňující navrhovat vzhled jednotlivých herních objektů.

Vypsané schopnosti souvisí s vývojem hry jako takovým. Ovšem dobrý herní designér musí mít i jiné schopnosti. Například musí umět pracovat v týmu, což zahrnuje schopnost komunikace, brainstormingu nebo veřejného vystupování kvůli prezentování svých nápadů.

Ovšem naprosto nejdůležitější schopnost zatím zmíněna nebyla a tou je schopnost naslouchat. Tím není myšlena schopnost slyšet něčí slova, nýbrž schopnost pochopit a odvodit myšlenky případně požadavky z vyřčených slov. Naslouchat je třeba v první řadě ostatním členům vývojovému týmu, konzultovat jejich názory a nápady. Dále je třeba naslouchat cílové skupině a tedy požadavkům cílových uživatelů a klientovi, tedy osobě, která poptává návrh hry. Naslouchat je třeba i hře samotné, což znamená nalezení chyb, případně identifikace možností na zlepšení. V poslední řadě je nutné naslouchat sám sobě a vlastnímu kreativnímu myšlení [5].

2.3 Herní zážitek

Samotná hra je bezcenná dokud ji lidé nezačnou používat. Primární starostí herního designéra tedy není hra jako taková, nýbrž hráč a jeho herní zážitek. Za zážitek lze považovat téměř cokoli, od objektů co vidíme, přes aktivity, které provádíme až po pocity, jež nás naplňují. Výsledná hra není herním zážitkem, nýbrž prostředkem pro ztvárnění herních zážitků. Ty ani nemusí být prezentovány pomocí reálných objektů, což znamená, že výslednou vizualizaci lze z určité části ponechat na hráčově fantazii a představivosti.

Jedním z nástrojů jak analyzovat chování uživatele při prožívání herního zážitku je introspekce vlastního chování. Na základě svých myšlenek a pocitů lze usuzovat, co ve hře funguje správně a co ne. To ovšem přináší dvě zásadní rizika. Prvním je eventuelní možnost zkresleného vnímání reality. To přichází například ve chvíli nedostatečného porozumění dané problematice. Herní designér ovšem není vědec a stará se především o herní zážitek a ne o pravdivé ztvárnění reality. Není tedy nezbytně nutné toto riziko eliminovat. Druhým je fakt, že lidé mají rozdílný vkus a vnímají určité herní zážitky různým způsobem. To znamená, že pokud bude designér tvořit hru jen podle své introspekce, výsledek nemusí zaujmout nikoho jiného než lidi s podobným vkusem. Způsob jak se tohoto rizika vyvarovat je již zmíněné naslouchání ostatním zainteresovaným osobám [5].

2.4 Herní elementy

Každá hra obsahuje čtyři základní herní elementy. Žádný z nich není méně důležitý než ostatní a všechny se z velké části vzájemně ovlivňují. Mají podstatný efekt na herní zážitek a je tedy třeba všechny zohlednit, sladit a vyvážit v závislosti na typu vyvíjené hry. Jsou jimi [5]:

- Herní mechaniky: Každá hra má své postupy a pravidla. Herní mechaniky popisují cíl dané hry a možnosti jak tohoto cíle dosáhnout. Hráč má pomocí nich jasně definováno, co může a nemůže ve hře provádět.
- Příběh: V herním prostředí jde o sekvenci událostí, jež se v průběhu hry rozvíjí. Může mít lineární charakter, nebo se větvit, případně se dynamicky vytvářet.

- Estetika: Estetika je velmi důležitý element, jelikož je pro hráče nejviditelnější. Udává vzhled, zvuk a atmosféru hry.
- Technologie: Zvolená technologie pro vývoj hry určuje prvky, které je možné do hry přidat.

2.5 Tématika

Již první pohled na hru by měl naznačit, o co ve hře jde. Všechny prvky ve hře, ať už jde o grafické objekty nebo hudbu, by měly pasovat jeden k druhému a vytvářet tak jednotnou tematiku. Takto zasazená hra může u hráče vyvolat mnohem hlubší herní zážitek a navodit autentičtější atmosféru [5].

2.6 Mentální schopnosti hráče

Samotný herní zážitek může vzniknout pouze na jediném místě, a to v hráčově mozku. Hráč s ním pracuje pomocí svých čtyř základních mentálních schopností, konkrétně schopností modelování, soustředění, empatie a představivosti. Právě díky nim je hráč schopen u hry setrvat a zůstat motivován k jejímu hraní [5].

2.6.1 Modelování

Realita je příliš komplexní i pro lidský mozek, aby jí pochopil. Proto si modeluje realitu a zjednodušuje její aspekty tak, aby jí pochopil. Často si to ani neuvědomíme. Například pokud se díváme na duhu na obloze, vnímáme ji jako souběžné pruhy vzájemně odlišných barev. Skutečnost je taková, že zde není žádná ostrá separace barev, ale hladký přechod vlnových délek. Tyto barvy jsou tedy jen iluze a ne součást reality. Ovšem jde o velmi užitečný model reality.

Mentální modely hráče jsou velmi důležité pro herního designéra. Hra se zjednodušenými principy oproti realitě tvoří snadno vstřebatelný a manipulovatelný mentální model. Právě proto jsou hry vnímány jako relaxace. Oproti složitým principům a pravidlům reality má při hraní her hráčův mozek daleko méně práce.

2.6.2 Soustředění

Důležitou technikou, kterou náš mozek používá k pochopení okolního světa, je schopnost selektivně soustředit svou pozornost na určité objekty a filtrovat jiné. Díky tomu se například dokážeme v plné místnosti vzájemně komunikujících lidí soustředit pouze na vlastní konverzaci. Aby hráč neztrácel pozornost při hraní hry, je důležité do hry začlenit tyto body:

- Jasně definovat cíle: Pokud má hráč cíl, a ví, co má provést k jeho dosažení, je pro něj snadnější zůstat soustředěný na jeho dosažení.
- Podávat přímou zpětnou vazbu: Pokud hráč musí po provedení určité akce dlouho čekat na její efekt, může snadno ztratit pozornost.
- Neustále vytvářet výzvy: Pokud je lidská mysl podrobena výzvě, zůstává soustředěná. Hráč si ovšem musí myslet, že je schopen tuto výzvu splnit a zároveň se nesmí při plnění výzvy nudit z důvodu její jednoduchosti. Vzniká zde nutnost vyvážení obtížnosti hry s ohledem na hráčovy herní dovednosti. Ty se obvykle s přibývajícím časem stráveným ve hře zlepšují, tudíž je nutné zvyšovat i obtížnost.

2.6.3 Empatie

Jako lidské bytosti máme výbornou schopnost vžít se do prožitků, emocí a citění jiných. Díky mentálním modelům, které náš mozek při této příležitosti vytváří, umíme prožít emoce, i když je cílová osoba ve skutečnosti vůbec neprožívá. Herní design by měl být tvořen tak, aby byl hráč schopen sympatizovat s ovládanou postavou, pokud je ve hře přímo ztvárněna, případně aby jí hráčův mozek sám vytvořil pomocí mentálního modelu. Podstata her je totiž o řešení problémů a o rozhodování, a pokud se do hry vžije, půjde mu to podstatně lépe.

2.6.4 Představivost

Hráč nemusí být kreativní umělec, aby si uměl něco představit. Lidský mozek každého člověka si dokáže i o velmi malé informaci vytvořit její obrázek a doplnit zbytek sám. V případě změny prvotní informace dokáže ve velmi krátkém čase svou představu pozměnit.

2.7 Herní design strategické hry

Strategická hra je jedním z běžných herních žánrů, do kterého spadá i aplikace popisovaná v této práci. Hra tohoto žánru je charakterizována těmito znaky [6]:

- Dlouhodobé cíle: Oproti jiným herním žánrům trvá splnění cíle strategické hry obvykle déle a postup k jeho splnění je otevřenější.
- Plánování a taktika: Hra by měla hráči dopřát možnost takticky plánovat další postup, následkem čehož by plnil cíle efektivněji.
- Minimální dopad specifických akcí hráče na výsledek hry: Pokud hráč udělá dobrý či špatný dílčí tah, neznamená to, že automaticky vyhrál nebo prohrál hru.

2.7.1 Zásady při tvorbě designu

Odlišení dobré hry od špatné je subjektivní. Každý hráč je totiž jedinečný a preferuje něco jiného. Obecně lze ovšem uvést několik zásad, které lze uplatnit na jakéhokoliv hráče.

Možnost rozhodování je principem každé hry. Pokud jsou ale hráči nabízeny možnosti, kde je vždy patrné, která z nich je nejlepší, je dobré, je vůbec neprezentovat. Na takovou volbu nemusí hráč vynaložit žádné úsilí a volba tedy nemá ke hře žádný strategický přínos. Naopak hra by hráče měla vybízet k netriviálním rozhodnutím a rovněž ho obeznámit s následky jednotlivých možností. Hra by tedy neměla skrývat své pravidla a principy, nýbrž by je hráči měla prezentovat tak, aby je pochopil.

Přidáváním alternativních možností rozhodování je možno dosáhnout lepšího herního zážitku. To ovšem nemusí platit pro přidávání dalších herních objektů, respektive proměnných. Pokud jich je obsaženo hodně, je pro hráče těžší pochopit principy hry a může být frustrován. Každá možnost rozhodnutí by měla představovat určitou míru rizika. Tu by měl hráč při plánování dalšího postupu zohlednit. Malé riziko obvykle pro hráče znamená malou odměnu, naopak pokud podstoupí velké riziko, tak bude v případě úspěchu odměněn daleko více [6].

3 Stávající systémy pro výuku podnikání

Kapitola shrnuje aktuálně dostupné simulační systémy určené k výuce podnikání a to jak české, tak zahraniční. Zhodnocen bude zejména princip, uživatelské rozhraní a užitečnost aplikace pro začínající případně budoucí podnikatele.

3.1 Market Hero

Online webová simulační hra "Market Hero" vznikla na území České a Slovenské republiky. Na vývoji se podíleli vývojáři Fakulty elektroniky a informatiky VŠB-TUO a grafici Fakulty Riadenia a informatiky Žilinské univerzity.

Hra je konstruována tak, aby zapadala do paradigmatu výuky na středních školách. Je tedy nutná přítomnost učitele / administrátora, který bude dohlížet na průběh konkrétních instancí hry a případně je do určité míry ovlivňovat. Bez předem vytvořených instancí, není možné, aby se uživatelé do hry přihlásili. Aplikace tedy není cílená pro každého uživatele v prostředí internetu, ale pouze pro vybrané skupiny, kde spadají, jak již bylo zmíněno, ekonomické třídy na středních případně vysokých školách.

Uživatel musí při vstupu do hry nejprve založit společnost. Je mu zobrazen jednoduchý formulář obsahující vstupní pole pro název firmy, její popis a právní formu.

The screenshot displays the 'Market Hero 2' web application. At the top, there is a navigation bar with links: Domů, Hrát, Profil uživatele, Nastavení hry, Administrace, and Podpora. Below the navigation bar, the main content area is titled 'Seznam firem' (Company List). On the left side, there are three informational sections with icons and links to 'Dozvědět se více...' (Learn more...):

- Section 1: 'V podnikání byste se měli řídit několika základními pravidly, abyste zvýšili svou šanci uspět na trhu.' (In business, you should follow several basic rules to increase your chance of success in the market.)
- Section 2: 'Jak začít při podnikání a jak si vybrat nejvhodnější oblast zjistište v této sekci.' (How to start in business and how to choose the most suitable area will be clarified in this section.)
- Section 3: 'Oblast pro seznámení s ekonomickými ukazateli. Jejich vysvětlení a optimální využití pro firmu.' (Area for getting acquainted with economic indicators. Their explanation and optimal use for the company.)

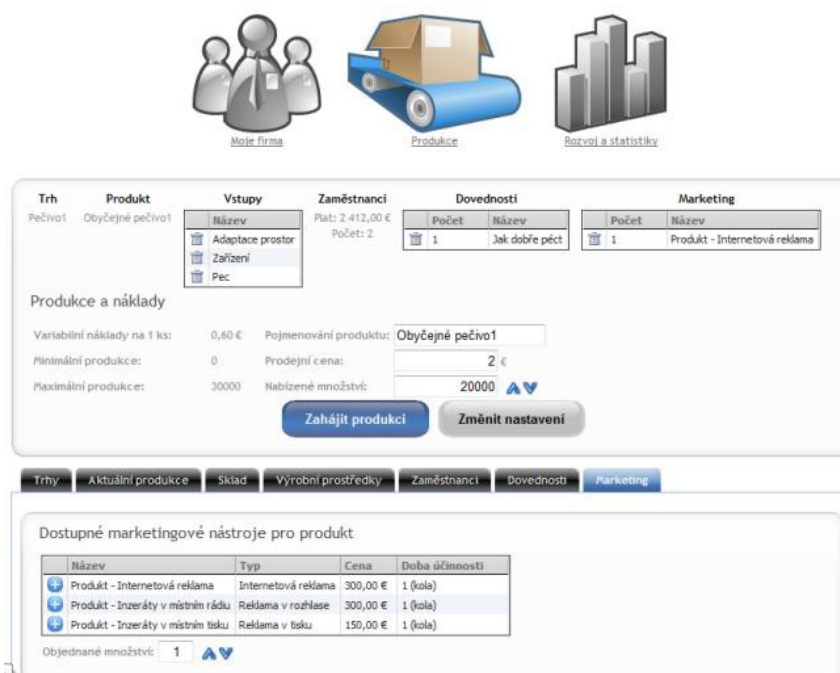
The main content area contains a list of companies. The first company is 'Jitka', owned by 'Jitka-pokusná' (Jitka-experimental), with the role of 'Vlastník' (Owner). Below the list, there is a 'Vytvořit firmu' (Create company) button. The form for creating a company includes fields for 'Název firmy:' (Company name), 'Popis firmy:' (Company description), and 'Právní forma:' (Legal form), which is currently set to 'Společnost s ručením omezeným' (Limited liability company).

	Název	Popis	Funkce
Vybrat	Jitka	Jitka-pokusná	Vlastník

Obrázek 3.1 – Založení firmy v aplikaci Market Hero

Zdroj: <http://markethero.eu>

Dále má možnost nahrát logo a vybrat spoluhráče, kteří budou moct kooperativně zasahovat do chodu firmy. Progres ve hře je řízen pomocí tahů. Uživatelé tak v každém tahu, dle svého uvážení a zvolené strategie, nastaví potřebné údaje a po odehrání tahu jsou prezentovány výsledky. Učí se tak reagovat na aktuální situaci a uvidí důsledky svých rozhodnutí v závislosti na konkurenci, tedy na ostatních uživateli. Aby tato konkurenceschopnost mezi uživateli vznikala, je sortiment prodáváných produktů značně omezen. Uživatelé ve hře řídí jak samotný prodej produktu, což obnáší stanovování prodejní ceny, volbu cílového trhu, zahájení marketingových kampaní a podobně, tak i jeho výrobu, tedy nákup výrobních prostředků, najímání a školení zaměstnanců a stanovování vyráběného množství. Vybavováním firmy vznikají fixní a variabilní náklady, které musí být uhrazeny. Uživatel tedy musí řídit své finanční prostředky s rozvahou, ovšem v krajním případě má možnost využít půjčku od banky.



Obrázek 3.2 – Řízení firmy v aplikaci Market Hero

Zdroj: <http://markethero.eu>

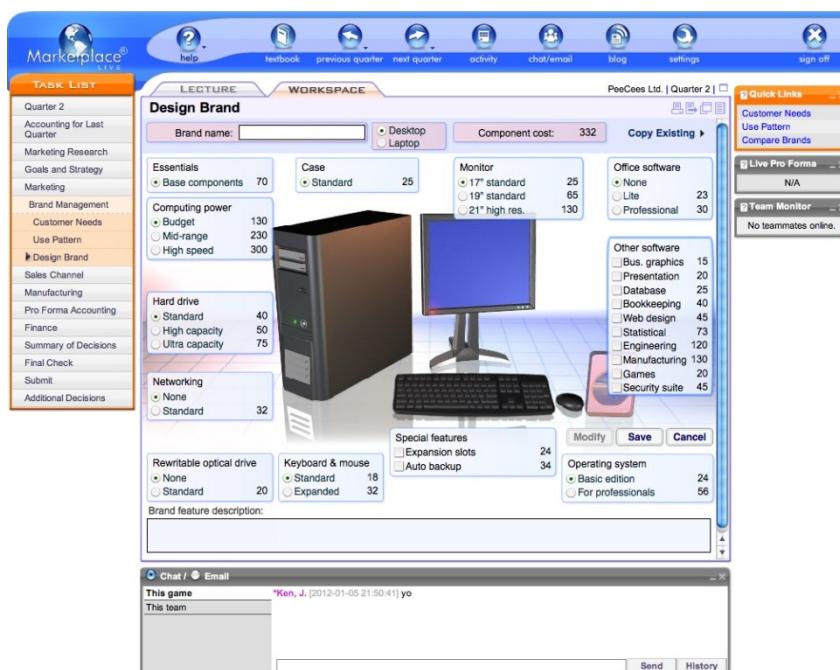
Aplikace je po grafické stránce dobře zpracována. Základní rozcestník v konkrétní instanci hry tvoří záložky "Moje firma", "Produkce" a "Rozvoj a statistiky". Řízení firmy pak znamená jednoduché vyplňování formulářů a výběr různých prvků z nabídky. Lze tedy říci, že je aplikace přehledná, ovšem pouhé vyplňování dotazníků není dostatečně zábavné pro mladé lidi, pro které je tato aplikace primárně určená. Aplikace po nich vyžaduje velkou představivost, aby dokázali aproximovat rozhodnutí, týkající se chodu firmy v této virtuální simulaci na rozhodnutí v reálném světě [7].

3.2 Marketplace Live

Simulační prostředí Marketplace Live bylo vyvinuto společností Innovative Learning Solutions, Inc. sídlící ve Spojených státech Amerických. Je použit robustní simulační model, který byl společností optimalizován již přes 20 let.

Stejně jako aplikace Market Hero je i zde předpokládána existence učebních skupin a jejich instruktorů. Kurzy jsou uzavřené a pro jejich užívání je nutné zakoupit licenci. Struktura kurzů je navržena pro konkrétní odvětví podnikatelské sféry sahající od obecnějších témat, jako jsou základy marketingu, business strategie přes management a logistiku až po konkrétnější témata například e-komerci a účetnictví. Uživatelé, dle typu kurzu, soupeří buď s počítačem ovládanou konkurencí, nebo sami mezi sebou.

Simulace probíhá po jednotlivých etapách (kvartálech), kdy uživatelé provádějí strategická rozhodnutí. Obsažen je celý životní proces prodáváného produktu od jeho návrhu, kde je možné v grafickém rozhraní volit jednotlivé komponenty a vlastnosti přes výrobu až po propagaci a prodej. Trh je rozdělen do několika segmentů jednak z geografického, a jednak z demografického pohledu. Ve vhodně prezentovaném grafickém rozhraní je tedy možné vybírat oblasti (města), kde bude firma působit, a v rámci nich vybírat cílové skupiny lidí s určitými vlastnostmi a touhami, jež se vážou k typu navrženého produktu, například při nákupu počítače se lidé zajímají o jeho výkon, velikost obrazovky apod. Pro jednotlivé oblasti je možné najmout prodejce a vyškolit je pro prodej pouze určitým segmentům lidí.



Obrázek 3.3 - Ukázka aplikace Market Hero Live

Zdroj: <http://www.marketplace-simulation.com/>

Prostředí aplikace je ve velké míře ztvárněno pomocí 3D grafiky, která je na vhodných místech provázána s uživatelem zadanými vstupními údaji ve formulářových prvcích. Například při navrhování produktu a kombinování patřičných komponent, se mění jeho vzhled a výsledek je uživateli prezentován jako trojrozměrný model, nebo při řízení výrobního procesu, se u výrobní linky zobrazí postavy zaměstnanců v závislosti na uživatelem určeném počtu [8].

3.3 SimVenture

Na rozdíl od zatím popsaných aplikací není aplikace SimVenture provozována online a nelze s ní pracovat prostřednictvím webového prohlížeče, nýbrž je nutná instalace na klientské počítače. Software byl vyvinut Britskou společností Venture Simulations Ltd a uvolněn v roce 2006.

Aplikace se zaměřuje na individuální výuku jednotlivých uživatelů případně malých skupin uživatelů (2 - 4) a chybí zde možnost simulace ve sdíleném prostředí a tedy i možnost organizování výuky do studijních skupin.

Neomezené užívání aplikace je podmíněno zakoupením licencí, nicméně pro vyzkoušení je k dispozici demoverze, která oproti plné verzi limituje použití některých funkcí, jako je ukládání a načítání jednotlivých her, i časový rámec hry.

Základní navigaci v aplikaci tvoří přehledná struktura možností organizovaná do odvětví managementu organizace, financí, marketingu a produkce výrobků, jejichž pestrost je zde limitována pouze na počítače (viz Obrázek 3.4).

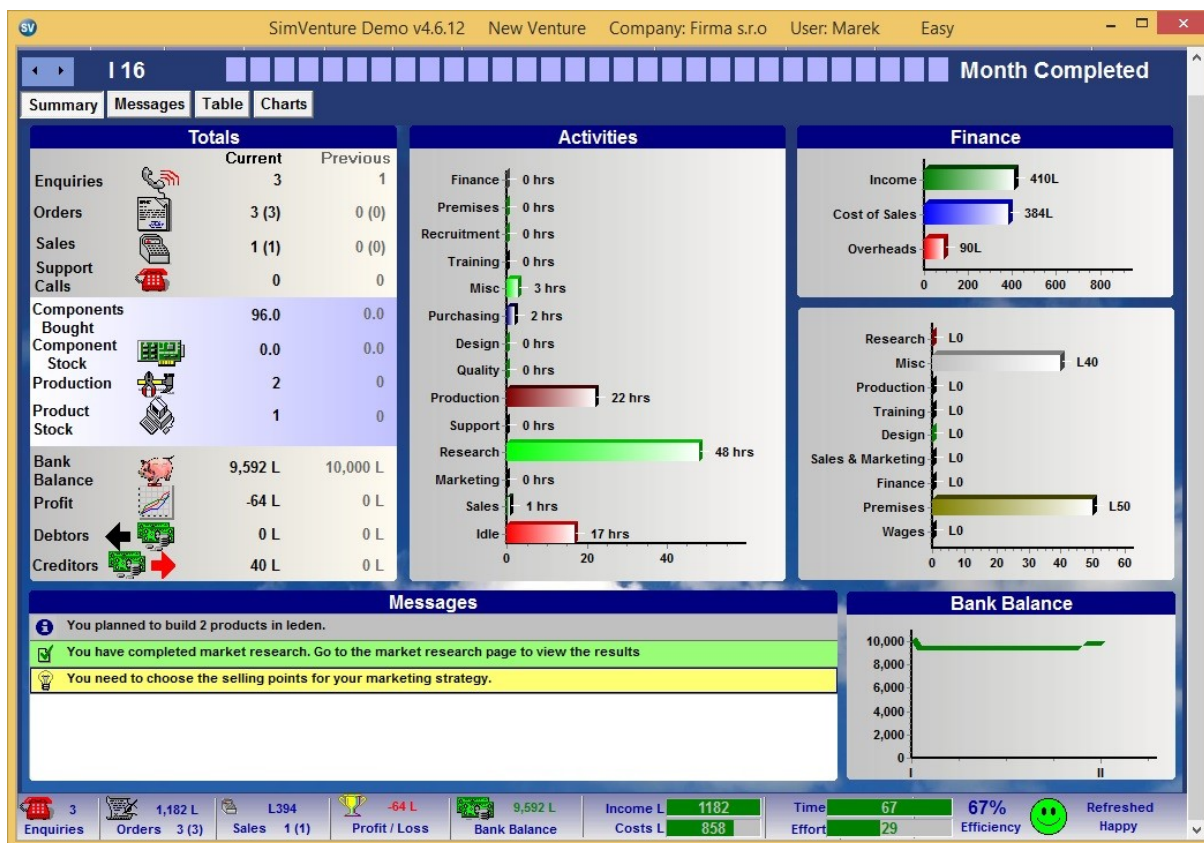


Obrázek 3.4 – Navigační menu v aplikaci SimVenture

Zdroj: SimVenture demoverze

Simulace probíhá formou tahů, které uživatel řídí samostatně. Odehrání jednoho tahu odpovídá z časového hlediska uplynutí jednoho měsíce herního času. Mezi tahy může uživatel

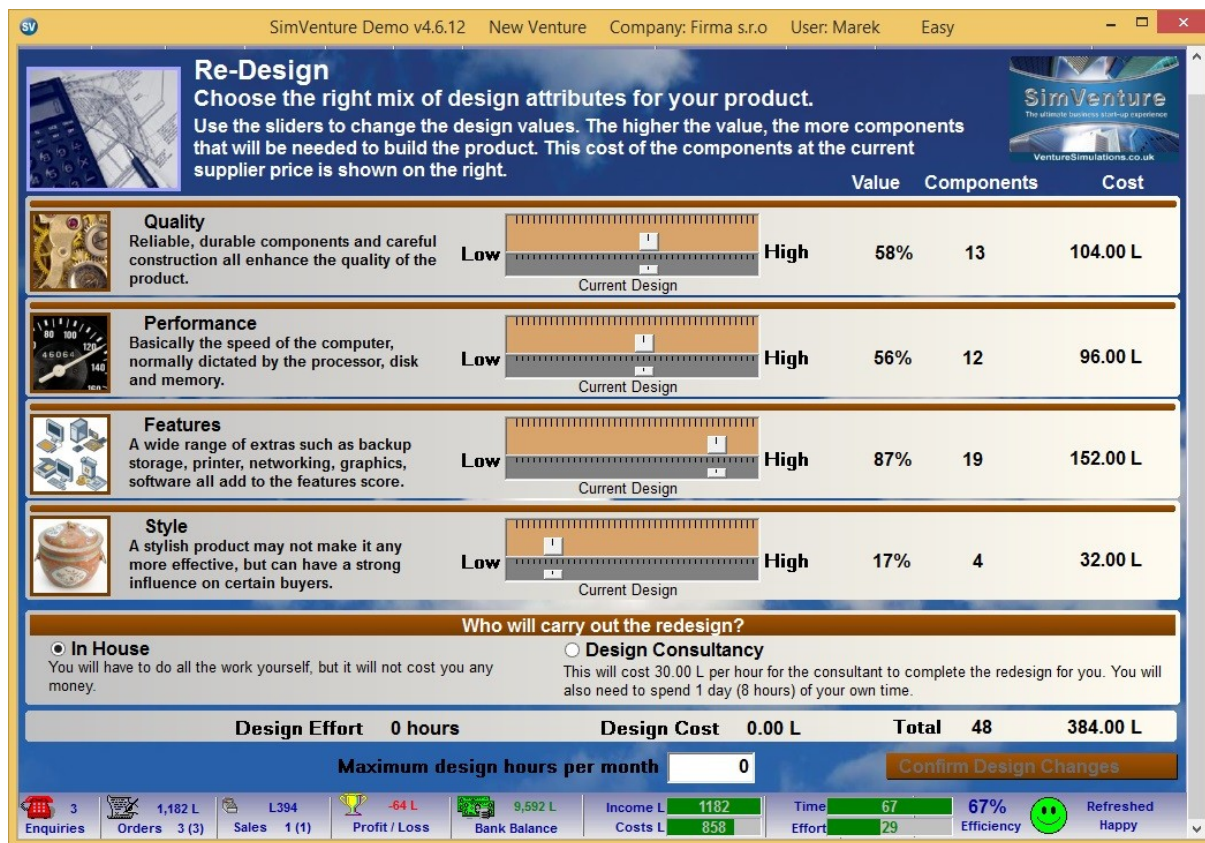
provádět strategická rozhodnutí, které spadají do zmíněných odvětví. Každé rozhodnutí je ve hře popsáno, vysvětleno a jsou zdůrazněny klíčové body, které by měl uživatel zohlednit. Většina z nich vytváří úkoly, které je nutné rozdělit mezi zaměstnance společnosti tedy i jejího majitele, který ve hře zastupuje uživatele. Provádění úkolu, ať už jde o průzkum trhu, výrobu produktů, nebo klientskou podporu a servis, stojí zaměstnanec čas, který je nutné v každém tahu rozdělovat s rozvahou. Po provedení tahu jsou uživateli prezentovány dosažené výsledky, provedené aktivity, finanční přehled a podobně (viz Obrázek 3.5), což by měl ve svých následných rozhodnutích zohlednit.



Obrázek 3.5 – Odehrání tahu v aplikaci SimVenture

Zdroj: SimVenture demoverze

Úspěch společnosti závisí, v první řadě, na schopnosti uživatele uspokojovat potřeby zákazníků a zohledňovat nabídku konkurence. Proto by měl uživatel pravidelně provádět průzkum trhu, průzkum konkurence a průzkum potřeb zákazníků, které po dokončení odhalí aktuální segmentaci trhu, nabídku a poptávku. Na základě těchto informací se může uživatel zaměřit na konkrétní segmenty zákazníků a uzpůsobit pro ně své produkty. Odlišnost segmentů je popsána odlišnou preferencí po kvalitě, výkonu, funkcionalitě a stylu produktu. Návrh produktů je potom zcela v režii uživatele a je jen na něm, na který segment daný produkt uzpůsobí. Čím bude zastoupení dané vlastnosti u produktu větší, tím vyšší budou náklady na jeho výrobu a tím vyšší by logicky měla být i prodejní cena, ovšem uživatel by měl zvážit, zda pro cílový segment není moc vysoká. Rozhraní pro návrh produktu je vidět na obrázku níže (Obrázek 3.6) [9].



Obrázek 3.6 – Návrh produktu v aplikaci SimVenture

Zdroj: SimVenture demoverze

3.4 Travian

Doposud byly analyzovány řešení obsahující převážně výukové principy. V dnešní době jsou ovšem na internetu zdarma k dispozici stovky až tisíce online strategických her, přičemž většina z nich obsahuje nějaký ekonomický systém a dovoluje například získávat a spotřebovávat různé zdroje nebo obchodovat s ostatními uživateli. Již kolem roku 2000 začaly podobné hry vznikat. V roce 2004 byla spuštěna v dnešní době jedna z nejznámější a nejpopulárnějších online her Travian.

Prostředí Travianu je zasazeno do období pozdní antiky a vystupují v něm národy Říma, Germánie a Galie. Každý z nich poskytuje lehce odlišný herní zážitek. Princip hry spočívá v budování, vylepšování a dobývání vesnic. Vše stojí na ekonomickém systému, ve kterém figurují čtyři základní suroviny, které mohou být získány jak vlastní těžbou, tak obchodem s ostatními uživateli, případně drancováním jejich vesnic.

Každá vesnice má dány své souřadnice v dvourozměrném prostoru, což dovoluje, jejich dynamické vykreslování ve dvourozměrné sdílené mapě (viz Obrázek 3.7). Mapa nezobrazuje celý herní svět, nýbrž jen jeho část, ovšem umožňuje interaktivní posun i mírné oddálení.



Obrázek 3.7 – Mapa vesnic ve hře Travian

Zdroj: <http://travian.cz>

Hra samozřejmě poskytuje i detailní pohled na konkrétní vlastní vesnici. Ten je rozdělen na blízké okolí vesnice a na její centrum. V okolí vesnice je možné navyšováním úrovně příslušných těžebních dolů navyšovat celkovou produkci zmíněných surovin. Centrum vesnice lze vybavovat různými budovami, kdy každá má svůj jasný účel. Například pro uskladnění získaných surovin je nutné postavit sklad, pro obchodování s ostatními uživateli tržiště, pro rekrutování vojenských jednotek zase kasárny a podobně. Budova může být postavena na jednu z několika dostupných stavebních parcel, přičemž některé typy mohou být umístěny vícekrát a uživatel tak má možnost ovlivňovat strategický účel dané vesnice (viz Obrázek 3.8).

Budování a vylepšování stojí určité množství surovin a také zabere určitý čas. Současně lze provádět maximálně 2 tyto operace, a před provedením další, je uživatel nucen čekat na dovršení časového odpočtu. Obdobný logický princip platí i pro přesun vojenských jednotek a pro přesun surovin.

Hra je pro uživatele dostupná zdarma, ovšem nabízí i placené služby. Ty zahrnují například možnost dočasné navýšení produkce určité suroviny, okamžité dokončení rozpracované stavby nebo i aktivaci komplexního zvýhodňujícího balíku, který rozhraní aplikace doplní o další užitečné funkce, poskytne doplňující výpisy atd. Služby je možné aktivovat pomocí speciální interní měny (Zlaté), kterou lze získat výměnou za reálné peníze.

Travian je nyní celosvětově používaná hra, běžící na několika serverech, na kterých jsou potom vytvářeny herní světy. Počet aktivních uživatelů je udáván v řádech statisíců až milionů. V České Republice je tento počet kolem devíti tisíc [10].



Obrázek 3.8 – Centrum vesnice ve hře Travian

Zdroj: <http://travian.cz>

4 Vize vlastního řešení

Na základě studia existujících aplikací a obecně známých herních a ovládacích principů, bude v této kapitole ztvárněna představa o vlastním navrženém řešení problematiky simulace podnikání.

Na rozdíl od většiny existujících systémů, se tato aplikace nebude přímo zaměřovat na výuku v klasickém schématu s učitelem a studenty. Vstup do aplikace tedy nebude podmíněn členstvím v určité skupině a bude se moct přihlásit kdokoli a kdykoliv, to znamená, i do probíhající hry. Vzdělávací obsah bude začleněn přímo do aplikace a bude zjednodušen pouze na vysvětlení použitých pojmů v kontextu uživatelem prováděné akce.

Předpokládaným cílovým segmentem jsou lidé s oblibou strategických her, ale také lidé, kteří chtějí v blízké době začít podnikat, nebo právě začali. Segment tvoří ve velké míře muži a ženy studující střední nebo vysokou školu a bude to bráno v úvahu při návrhu uživatelského rozhraní. Studenti ekonomických oborů se v aplikaci zorientují mnohem lépe, ale je počítáno i s účastí studentů, zaměřujících se na jiné profese. Potenciálním klientem je i ekonomická škola jako taková, která by tak mohla svým studentům nabízet obohacující interaktivní způsob výuky. Aplikace nebude prezentovat scénář, kdy má podnikatel k dispozici velký kapitál a může si ihned dovolit provádět rozhodnutí na úrovni středních a velkých firem. Naopak bude uživatelům ukazovat, že založit a vést firmu může i obyčejný člověk a to třeba i z vlastního domova a bez zaměstnanců. S tímto scénářem se většina lidí cílového segmentu ztotožní mnohem lépe.

Průběh simulace bude z časového hlediska probíhat tahově. Všechny provedené změny, se tedy neprojeví ihned, ale až při odehrání následujícího tahu. Aplikace bude tahy provádět automaticky v určitých časových intervalech. Ty by neměly být moc dlouhé, aby nebyl uživatel vystavován dlouhému čekání na změny, ale ani ne moc krátké, aby nedocházelo k poklesu výkonu.

Aplikace nebude tvořena jen jednou hrou respektive herní instancí, nýbrž jich může být vytvořeno více, přičemž jejich funkčnost bude vzájemně nezávislá a budou moct pracovat i s rozdílnými hardwarovými prostředky. Instance budou oprávněnou osobou do jisté míry konfigurovatelné a jejich odlišnosti budou znát jak z grafického tak z funkčního pohledu. Počet uživatelů registrovaných ke konkrétní hře, bude moct být z důvodu udržení přijatelného výkonu omezen.

Představená aplikace je značně rozsáhlá a bude podrobněji popisována po jednotlivých blocích.

4.1 Postava podnikatele

V každé herní instanci bude uživatele zastupovat jeho herní charakter. Při prvním vstupu do hry bude tento charakter vytvořen. Ve vytvořených společnostech pak může vystupovat jako klasický zaměstnanec, který ovšem nebude dostávat výplatu.

4.2 Mapa prostředí

Nejglobálnější navigace v aplikaci bude vyobrazena pomocí libovolných obrázků, jež budou představovat herní prostředí, ve kterém budou začleněny body, představující v geografickém pojetí lokální trhy. Dále budou tyto body označovány jako regiony.

Oprávněná osoba bude mít možnost mapu upravovat a bude tedy moci definovat jak obrázkové podklady a jejich umístění, tak i vlastnosti a umístění regionů.

Mapa nebude ukončena okraji obrazovky, z čehož vyplývá nutnost implementace posunu, případně přibližování mapy. Samozřejmostí je i možnost interakce s jednotlivými regiony a to jak pro editační účely, tak pro účely navigace, kdy dojde k přesměrování uživatele na mapu konkrétního regionu.

4.3 Mapa regionu

Uživatelské rozhraní by mělo vizuálně připomínat strategickou hru. Regiony tedy budou ztvárněny pomocí mapy, jež bude tvořena obrázky s izometrickým zobrazením budov a jiných struktur. Podobně jako u mapy prostředí, i zde nebude mapa ukončena a budou zde očekávány podobné možnosti interakce. Uživatelské akce pro konkrétní budovu se ale budou lišit v závislosti na typu budovy, kterých bude v aplikaci definováno několik.

4.3.1 Budovy

Budovy budou v aplikaci vizuálně rozeznatelné a každá tedy bude vyobrazena pomocí vlastního obrázku. Jak bylo zmíněno výše, budou spadat mezi několik typů. Zde je jejich stručná charakteristika:

- **Banka:** Pokud bude uživatel potřebovat finanční prostředky, může využít služeb banky, která mu poskytne úvěr. Uživatel si sám zvolí výši úvěru a dobu splácení a z těchto údajů mu bude vypočítána výše splátky za každý následující tah. Ke konkrétní instanci banky se vztahuje hodnota úrokové míry a hodnota maximální výše úvěru.
- **Radnice:** Pro správu svých společností, případně pro prohlížení ostatních společností, musí uživatel využívat tuto budovu. Vytvoření společnosti bude podmíněno zaplacením správního poplatku, přičemž jeho výše bude podléhat konkrétní instanci radnice.
- **Úřad práce:** Seznam nezaměstnaných uchazečů o práci v daném regionu bude dostupný, právě po navštívení této budovy. Uživatel má možnost procházet jejich životopisy, kde uvidí, jak moc se tento potenciální zaměstnanec hodí k vykonávání požadované práce. Právě zde může uživatel daného zaměstnance najmout případně propustit.
- **Dekorace:** Tyto struktury nemají žádný zvláštní účel a v aplikaci budou sloužit pouze pro zkrášlení herního prostředí. Můžou zde spadat například silnice, chodníky, stromy apod.
- **Veřejné budovy:** Aby měl uživatel kde provozovat své podnikání, bude muset navštívit tyto budovy. Každá z nich obsahuje jednotky nemovitostí, které si uživatel může zakoupit. To znamená, že si například nebude moci koupit panelový dům, ale pouze byt. Tyto budovy se budou vizuálně řadit do několika typů od panelových domů, přes nákupní centra, až po sklady a výrobní haly. Toto rozdělení ovšem nebude uživateli striktně nařizovat, k jakému účelu bude budova sloužit, a aplikace uživateli dovolí, si například v bytě vystavět prodejní prostory. Ovšem z pohledu reálného světa je pravděpodobnost, že zákazník navštíví prodejnu vystavěnou v bytě podstatně menší než prodejnu vystavěnou v obchodním centru. Z tohoto důvodu bude předdefinován koeficient návštěvnosti pro každý typ veřejné budovy.

Oprávněná osoba bude moct tyto města upravovat, což zahrnuje umísťování nových budov, případně jejich přesun, otáčení a odebírání. Umístění budovy na novou pozici bude validní, pouze v případě, pokud do nové pozice nebude zasahovat jiná struktura než ta, která určuje podklad budovy, což může být například tráva. Budovy, které nemají definován potřebný podklad, půjde umístit kamkoliv.

Veřejné budovy pak budou představovat výchozí navigační bod k přesměrování na mapu konkrétní nemovitosti.

4.4 Mapa nemovitosti

I tahle mapa bude obdobně jako mapa regionu izometrického zobrazení. Ovšem budovy a venkovní struktury zde budou nahrazeny vybavením a místnostmi.

4.4.1 Vybavení nemovitosti

V aplikaci bude dostupných několik typů vybavení, kdy každé bude mít svůj daný účel a bude ovlivňovat chod společnosti. Zde je uveden základní přehled:

- Návrhová tabule: Bude sloužit k vytváření prototypů produktu a tedy k definování sortimentu, se kterým společnost operuje. Počet prototypů bude limitován, nicméně může být navýšen zakoupením dalších návrhových tabulí.
- Prodejní regál: Vybavení nezbytné pro prodej zboží. Prodejní regál bude mít definovanou svou kapacitu a maximální počet typů vystavovaných produktů. Uživatel pak bude tyto produkty vybírat ze svého dostupného sortimentu a stanovovat cenu. Aktuální množství vystavených kusů se bude snižovat s přibývajícím počtem objednávek a musí být doplňováno.
- Skladovací regál: Bude sloužit pouze pro skladování produktů, které nejsou naskladněny v prodejních regálech a bude mít zpravidla větší kapacitu.
- Pokladna: Bez pokladny není možné realizovat žádné prodeje. Na pokladně bude dostupný přehled již realizovaných objednávek i objednávek čekajících na namarkování.
- Konstrukční stůl: Zde bude probíhat produkce požadovaných produktů v požadovaném množství.
- Pracovní stůl: V aplikaci nebude mít významnější roli, bude pouze poskytovat přístup k záznamům o příjmech a výdajích, což bude simulovat nahlížení do účetnictví společnosti.
- Automobil: Po zakoupení více provozoven společnosti budou automobily zajišťovat transport mezi nimi.

4.4.2 Standardní vybavení v budově

Oprávněná osoba bude moct definovat standardní vybavení pro konkrétní instanci veřejné budovy. Bude tak vytvořena pomyslná šablona, která se použije pro všechny existující nemovitosti v dané budově.

Kromě umístění výše zmíněných typů vybavení bude definice obnášet i určení podkladů, omezující možnosti umísťování. Ty budou dále označovány jako místnosti, přičemž v aplikaci budou taktéž členěny do několika instancí, kdy každá z nich pojme různé typy vybavení. Poměrem velikostí

místností lze nepřímo ovlivňovat účel dané veřejné budovy, to znamená například navýšením prostoru pro skladování a snížením, eventuálně i vynecháním, prostoru pro prodejní akce, vznikne sklad.

4.4.3 **Nákup a prodej vybavení**

Zakoupením nemovitosti dojde k automatickému zakoupení předdefinovaného standardního vybavení, což bude mít vliv na celkové pořizovací náklady. Zakoupení dalšího vybavení, případně jeho prodej nebo jen jeho přesun, je pak zcela na uživateli. Bude limitován pouze finančními možnostmi a volným prostorem v dané místnosti.

4.5 **Živnostenské oprávnění**

Před zahájením podnikání je nutné získat živnostenské oprávnění. V plánované aplikaci budou dostupné podnikatelské činnosti omezeny pouze na výrobu distribuci a prodej zboží a nebude možné poskytovat zákazníkům služby. Výběr tedy spadá pouze do kategorie volných živností.

Každý uživatel bude mít možnost spravovat více živnostenských oprávnění. U každé bude moct z nemovitostí, kde je vlastníkem, zvolit ty, které budou představovat provozovny společnosti, přičemž každá bude moct být provozovnou pouze v jedné.

V aplikaci bude možnost podnikat jako fyzická osoba pouze na živnostenské oprávnění nebo jako právnická osoba ve společnosti s ručením omezeným (s. r. o.). Každé vzniklé živnostenské oprávnění bude mít, spolu se samotnou postavou podnikatele, k dispozici finanční rozpočet, ze kterého může čerpat. Všechna vzniklá oprávnění k podnikání fyzické osoby, budou, včetně postavy podnikatele, sdílet stejný rozpočet. Naopak dojde-li k vytvoření oprávnění k podnikání právnické osoby, bude vytvořen nový rozpočet, do něhož bude přidán kapitál ve výši, kterou určí uživatel sám.

4.6 **Trh**

Všichni uživatelé budou v rámci jedné instance hry součástí trhu, který bude ovlivňován jejich rozhodováním při vedení firmy. Trh bude segmentován jak geografickými faktory, tak faktory demografickými a psychografickými.

4.6.1 **Kapacita trhu**

Pro udržení konkurenceschopnosti a zároveň udržení přiměřeného počtu zákazníků by měla být kapacita trhu nastavena s ohledem na počet uživatelů v konkrétní instanci hry. Kapacita trhu bude v aplikaci vyjádřena počtem zákazníků.

4.6.2 **Geografická segmentace trhu**

Jak již bylo zmíněno, trh bude rozdělen do graficky znázorněných regionů. V každém regionu bude uživatel pracovat s různým počtem potenciálních zákazníků a s jejich rozdílnými demografickými a psychografickými vlastnostmi a tedy i s rozdílnou poptávkou a nabídkou produktů.

4.6.3 **Demografická a psychografická segmentace trhu**

Oprávněná osoba bude moct dle svého uvážení určit faktory, kterými budou rozdělováni zákazníci a vytvořit tak konkrétní segmenty. Zákazníkům pak bude jeden z těchto segmentů přiřazen a budou tak určeny jeho zájmy i chování při nákupu produktů. Segmenty budou definovány pro celou

herní instanci, ovšem množství osob zastupující daný segment, bude moci být různý pro konkrétní regiony.

Společně se segmenty budou konfigurovatelné i globální zájmy zákazníků. Ty by měly mít alespoň nějakou návaznost na prodejné produkty v herní instanci. Kartézským součinem množin segmentů a zájmů vznikne dvourozměrná matice, kde každé pole bude udávat procentuální míru toho, jak moc má daný segment v oblibě daný zájem. Matice bude existovat v každém regionu nezávisle a bude v nich vyjadřovat aktuální trendy.

Tabulka 4.1 poskytuje ukázkou struktury této matice.

Zájem	Muž	Žena
Volání	45 %	50 %
Psaní zpráv	37 %	68 %
Poslech hudby	70 %	52 %

Tabulka 4.1 – Ukázka matice trendů

4.6.4 Průzkum trhu a průzkum konkurence

Matici trendů pro každý region bude mít každý uživatel k dispozici. Hodnoty uvidí vždy v aktuálním stavu. Kromě trendů mu budou prezentovány i informace o populaci včetně zastoupení segmentů v regionu a informace o prodejnosti produktů dle jednotlivých kategorií. S těmito znalostmi pak má šanci získat lepší pozici na trhu, pokud bude na změny reagovat a přizpůsobovat svůj sortiment.

Uživatel bude moci navštěvovat konkurenční společnosti a jejich pobočky. Infrastruktura společnosti mu bude skryta. Zobrazí se pouze pokladny a prodejní regály s vystavenými produkty, což je vše co obyčejný zákazník potřebuje k nákupu.

4.7 Produkty

Prodej produktů bude v aplikaci jediná možnost finančního příjmu. Produkty budou fiktivní a nebudou mít žádnou vazbu na skutečný trh. Podobně jako v každém internetovém obchodě, i v této aplikaci budou řazeny do struktury kategorií a prezentovány pomocí stručného popisu a fotografií.

Popisovaná hierarchická struktura bude upravitelná, ale pouze oprávněnou osobou v administračním rozhraní. Uživatelé si tedy nebudou moci v rámci herní instance vytvářet vlastní produkty, aby nedocházelo k nevyváženosti. Budou moci využít pouze ty předdefinované.

4.7.1 Parametry produktu

Aby byly produkty rozlišitelné nejen svým vzhledem ale i charakterem, budou v aplikaci popsány pomocí několika parametrů. Ty budou nepřímo ovlivňovat jejich prodejnost v závislosti na aktuální konfiguraci trhu.

- Kvalita: Referenční hodnota udávající jak moc je produkt kvalitní a tedy jak moc s ním bude zákazník spokojen. Ten se ale bude dívat i na cenu, za kterou je daný produkt danou společností nabízen a posoudí tak celkový poměr mezi kvalitou a cenou.
- Trvanlivost: Tato hodnota bude udávat dobu po jakou je zákazník s produktem spokojen a nebude si chtít pořízovat nový z dané kategorie. Doba bude v tomto případě znamenat počet uskutečněných tahů.
- Cílení: Obdobně jako se k regionu vztahuje matice vyjadřující trendy, se podobná matice bude vztahovat k produktu. Ta bude vyjadřovat jak moc je produkt cílený na segment a na uspokojení zájmů.
- Náкупní cena: Při výrobě a vytváření prototypu produktu se uživateli tato částka odečte z firemního rozpočtu.

4.7.2 Životní cyklus produktu ve společnosti

V rámci firemní infrastruktury bude práce s produktem procházet několika pomyslnými fázemi, než bude předán konkrétnímu zákazníkovi. Zde jsou vypsány v chronologickém pořadí:

1. Návrh produktu: Aby společnost mohla s produktem vůbec pracovat, bude nutné nejprve vyhotovit jeho návrh. Toto bude nutné absolvovat pouze jednou.
2. Výroba produktu: Po vyhotovení návrhu produktu, ho bude možné začít vyrábět, přičemž počet kusů k výrobě je na uživateli a na jeho finančních možnostech. Také musí zajistit prostor pro uskladnění produktů. Kusy, které budou nad kapacitní možnosti, budou zahozeny.
3. Transport produktu: Pokud bude chtít uživatel daný produkt prodávat, případně jen naskladnit jinde než na místě výroby, může produkt transportovat. V cílové destinaci ovšem musí rovněž zajistit dostatek skladovacího prostoru obdobně jako při výrobě produktu.
4. Doplnění produktu do prodejního regálu: Prodeje budou moci být realizovány pouze pro produkty vystavené v prodejních regálech. Pokud zákazník provede objednávku, je patřičný počet kusů z regálu odebrán. Bude na uživateli, aby vždy zajišťoval dostatečné množství k prodeji tak, aby byl zákazník spokojen.
5. Namarkování produktu: Před obdržáním zisku z prodeje, bude nutné ještě náležitě objednávkou namarkovat.

4.8 Simulace prodeje

Při každém simulovaném tahu proběhne mimo jiné i generování objednávek a jejich položek pro všechny existující společnosti. K objednávkám můžou být přiřazeny jen produkty z dostupného sortimentu firmy, který je dán osazením prodejních regálů.

Rozhodnutí, zda si zákazník daný produkt koupí, bude záviset na několika faktorech (*Factor*) a do jisté míry na náhodě, jelikož pro každého zákazníka bude generováno náhodné číslo (*random*),

a to potom použito při rozhodování. Čím vyšší bude hodnota faktoru, tím vyšší bude pravděpodobnost zakoupení. V rámci segmentů půjde navíc konfigurovat koeficienty důležitosti těchto faktorů (*Coef*). To znamená, že výsledná pravděpodobnost bude, za předpokladu stejných hodnot faktorů, pro různé segmenty nabývat jiných hodnot. Konkrétní zákazník zakoupí konkrétního produkt v případě platnosti níže uvedené nerovnice.

$$\frac{\sum_{i=1}^n Factor_i * Coef_i}{n} > random$$

Factor – Faktor ovlivňující prodejnost produktu

Coef – Koeficient důležitosti faktoru

random – Náhodné číslo z intervalu (0, 100)

n – počet faktorů

Rovnice 4.1 – Výpočet prodejnosti produktu

Proměnná *n* určuje počet faktorů. V aplikaci jich bude figurovat celkem pět. V následujícím textu budou popsány detailněji.

4.8.1 Návštěvnost veřejné budovy

V kapitole 4.3.1 byly popsány charakteristické znaky typů budov, přičemž u veřejných budov byl zmíněn koeficient návštěvnosti. Je to jeden z faktorů ovlivňující prodejnost. Jen díky správně zvolené budově pro vytvoření pobočky společnosti, bude tedy možno dosáhnout značného zisku.

4.8.2 Trendy

Pro vysokou hodnotu tohoto faktoru bude důležité správně zvolit produkty pro sortiment společnosti. Záleží totiž na tom, jak moc jsou v regionu populární segmenty a jejich zájmy, na které je produkt cílený. To znamená, že čím jsou rozdíly hodnot na náležitých pozicích v maticích trendů v regionu (*TrendFocus*) a cílení produktu (*ProductFocus*) menší, tím vyšší je hodnota tohoto faktoru. Na matici trendů mohou působit jisté externí vlivy, které ovlivní trendy na náležitých pozicích s ohledem na čas (*InfluenceFocus*). Administrátor bude moci těmto vlivům definovat opakující se časové úseky a to pro dny v roce, dny v týdnu a hodiny. Výpočet je uveden níže.

$$Factor = \frac{\sum_{i=1}^n 100 - |ProductFocus_i - TrendFocus_i + \sum_{j=1}^m InfluenceFocus_{ij}|}{n}$$

ProductFocus – hodnota cílení produktu na zájem

TrendFocus – hodnota trendu zájmu v regionu

InfluenceFocus – hodnota trendu zájmu v působícím vlivu

n – počet zájmů

m – počet působících vlivů

Rovnice 4.2 – Výpočet faktoru trendů

Po každém vyřízení objednávky se budou trendy v regionu dynamicky měnit a to tak, že se hodnoty na náležících pozicích přiblíží k hodnotám cílení produktu. Velikost této změny bude nepřímo záviset na celkovém počtu obyvatel v regionu. Pokud by tedy byl obyvatel pouze jeden, byly by změny stoprocentní.

4.8.3 Cena produktu

Nastavení ceny produktu bude zcela na uživateli, ovšem pokud chce, aby se produkt dobře prodával, musí se přizpůsobit vnímání zákazníka, ceně konkurence a zvážit své náklady. Všichni zákazníci budou mít společnou představu o tom, kolik by měl produkt v dané kategorii stát a jakou by měl mít kvalitu. Čím víc je stanovená cena v poměru ke kvalitě (*Ratio*) pod touto hranicí, tím větší je pravděpodobnost, že si zákazník produkt koupí. Naopak pokud bude nad hranicí, bude pravděpodobnost menší. Od této hranice se pak odvíjí horní (*RatioExpected_{Max}*) a dolní (*RatioExpected_{Min}*) mezní hodnoty, jež jsou o polovinu větší a menší. Ty poté určují stav, ve kterém nabývá výsledná pravděpodobnost stoprocentní a nulové hodnoty. Výpočet uvádí

Rovnice 4.3. Tato představa se nebude v aplikaci přímo vztahovat ke konkrétnímu produktu, nýbrž k náležícím kategoriím.

$$Ratio = \frac{PriceSelling}{Quality}$$

$$RatioExpected_{Min} = \frac{PriceSellingExpected}{QualityExpected} * \frac{1}{2}$$

$$RatioExpected_{Max} = \frac{PriceSellingExpected}{QualityExpected} * \frac{3}{2}$$

$$Factor = 100 - \frac{Ratio - RatioExpected_{Min} * 100}{RatioExpected_{Max} - RatioExpected_{Min}}$$

PriceSelling – Prodejní cena produktu

Quality – Kvalita produktu

PriceSellingExpected – Předpokládaná cena produktu

QualityExpected – Předpokládaná kvalita produktu

Rovnice 4.3 – Výpočet faktoru ceny produktu

Obdobně jako trendy se i představa o ceně bude dynamicky měnit s každou realizovanou objednávkou.

4.8.4 Poptávka zákazníka

Konkrétní zákazník bude vyžadovat konkrétní produkt, pokud ještě nevlastní žádný jiný z dané kategorie, respektive pokud vypršela doba, kdy je s produktem spokojen. Ta je určena parametrem trvanlivosti popsáným v kapitole 4.7.1. V takovém případě bude faktor nabývat maximální stoprocentní hodnoty, v opačném případě bude nulový.

4.8.5 Hodnocení zákazníka

Zákazník po první návštěvě pobočky společnosti provede její první hodnocení. To se pak s dalšími návštěvami bude upravovat. Zároveň se bude používat jako faktor určující prodejnost produktu.

Zákazník bude spokojený, pokud v pobočce najde produkt, který hledá a provede jeho nákup. Jeho hodnocení se navýší o předdefinovanou hodnotu, jež je určena segmentem zákazníka.

Pokud v rámci návštěvy pobočky stráví zákazník dlouhou dobu ve frontě u pokladny, bude jeho hodnocení naopak snižováno. To znamená, s každým tahem, kdy bude probíhat markování objednávky tohoto zákazníka. Obdobně jako u navyšování spokojenosti bude v segmentu předdefinována i hodnota určující o kolik se má spokojenost snížit.

Tento faktor hodně závisí na infrastruktuře společnosti. Uživatel musí zajistit, aby bylo vystaveno dostatečné množství produktů pro navyšování hodnocení zákazníka a aby u pokladen nevznikaly dlouhé fronty pro prevenci vzniku negativních hodnocení.

4.9 Zaměstnanci

Bez zaměstnanců nemůže fungovat žádná společnost. Nicméně aplikace bude prezentovat hlavně zahájení podnikatelské činnosti a její eventuální vzestup. Malou firmu může ze začátku podnikatel spravovat sám, a nepotřebuje další osoby. Proto bude v aplikaci reprezentující postava podnikatele (viz 4.1) zaměnitelná s ostatními zaměstnanci. Aplikace samozřejmě neumožní takovou osobu ze společnosti propustit, nebude ani možné jí nastavit výši finanční odměny. Na rozdíl od jiných zaměstnanců může figurovat ve všech vytvořených společnostech a jejich pobočkách.

4.9.1 Schopnosti zaměstnance

Po zaměstnání konkrétní osoby, bude třeba určit její náplň práce. U každého zaměstnance bude omezená kapacita jeho času. Ta bude udávána procentuálně a bude moct být rozdělována mezi různé aktivity u různého vybavení pobočky společnosti. Čím vyšší procentuální podíl uživatel nastaví, tím efektivněji bude práce vykonávána.

Každý zaměstnanec bude jedinečný a bude mít specifickou míru svých schopností, jež budou přímo souviset s dostupnými pracovními aktivitami. Schopnosti budou moct být obdobně jako v reálném životě dále rozvíjeny a vylepšovány. Bude tedy možné nalézt zaměstnance, který je vhodný pouze k určitým aktivitám. Ty ostatní samozřejmě může provádět taky, ale už ne tak efektivně.

K dokončení aktivity musí být vynaloženo určité úsilí. Aktivita se bude vztahovat k určitému subjektu, což může být například konkrétní instance produktu v případě výroby nebo konkrétní objednávka v případě markování zboží. Subjekt pak určí, kolik úsilí je třeba vynaložit. Přehled všech aktivit, potřebné vybavení a subjekty zobrazuje Tabulka 4.2.

Aktivita	Potřebné vybavení	Subjekt	Úsilí určuje
Návrh produktu	Návrhová tabule	Produkt	Kvalita produktu
Výroba produktu	Konstrukční stůl	Produkt	Kvalita produktu
Transport produktu	Automobil	Produkt	Vzdálenost cílové destinace
Doplňování zboží	Prodejní regál	Produkt	Objem produktu
Namarkování objednávky	Pokladna	Objednávka	Cena objednávky

Tabulka 4.2 – Aktivita

Vynakládáním úsilí získá zaměstnanec odpovídající počet zkušeností a s určitým počtem zkušeností bude automaticky povýšen na další úroveň. S každým povýšením získá zaměstnanec body, které pak může uživatel libovolně rozdělit mezi jeho schopnosti a tím je vylepšit. Počet těchto získaných bodů bude odpovídat počtu dostupných aktivit. Povýšování ani vylepšování schopnosti nebude možné provádět do nekonečna, nýbrž pouze do maximální hodnoty 100. Při dosažení maximální úrovně tedy bude mít uživatel dostatek bodů, aby je přerozdělil mezi schopnosti v maximální výši.

Zaměstnanci se stejně jako zákazníci budou vázat k určitému segmentu osob. Tomu půjde v administračním rozhraní také specifikovat jakým způsobem a jakou rychlostí bude zaměstnanec dosahovat dalších úrovní, pomocí specifikace počtu potřebných zkušeností (XP_{Level}) a koeficientu nárůstu těchto zkušeností s přibývající úrovní ($Factor$). Vztah pro výpočet množství zkušeností, potřebných pro postup na další úroveň uvádí Rovnice 4.4.

$$XP_{LevelUP} = XP_{Level} * (Level + 1)^{Factor}$$

XP_{Level} – Základní počet zkušeností k postupu na vyšší úroveň

$Level$ – Aktuální úroveň

$Factor$ – Koeficient nárůstu potřebných zkušeností s přibývající úrovní

Rovnice 4.4 – Výpočet množství zkušeností pro další úroveň zaměstnance

Mimo to půjde definovat rozsahy, ve kterých se budou pohybovat míry schopností u nově vygenerovaných osob.

4.9.2 Najímání zaměstnanců

Každý zaměstnanec bude moci být najat pouze jednou společností. Výběr zaměstnanců bude dostupný na úřadu práce a bude sdílen mezi všemi uživateli. Zobrazování budou pouze zaměstnanci z náležícího regionu. Po podepsání pracovní smlouvy a tedy po najmutí zaměstnance bude ihned z tohoto seznamu odebrán. Pokud bude zaměstnanec propuštěn nebo pokud podá výpověď, bude v seznamu opětovně dostupný.

Každý segment zaměstnanců bude mít svou představu o výši svého platu. Ta bude dále záviset na jeho úrovni pracovní vytiženosti. Čím vyšší úroveň zaměstnance a čím více je využívána kapacita jeho času, tím vyšší očekává plat.

Aby byl zaměstnanec ve společnosti spokojený, musí být dostatečně finančně oceněn. Nebudou-li splněny jeho podmínky, bude jeho spokojenost klesat. Při dosažení minimální úrovně se může stát, že dá zaměstnanec výpověď. Cílem uživatele tedy bude udržovat své zaměstnance spokojené a zároveň je neodměňovat příliš, aby společnost nezačala prodělávat.

5 Analýza požadavků

Tato kapitola vystihuje převod myšlenek vize na konkrétní požadavky navrhované aplikace. Je zde blíže specifikována funkční a datová stránka aplikace se snahou o systematický náhled.

5.1 Struktura systému

Výchozím bodem aplikace bude prostředí poskytující souhrnné informace o účelu systému doplněné o mediální prvky, tak aby nově přichodí uživatel okamžitě věděl co od systému očekávat. Zároveň půjde o rozcestník odkazující na jednotlivé instance probíhajících her, který bude adaptivně pracovat s uživateli a řídit jejich přístup na tyto hry. Instance her mohou být privátní, kdy bude přístup podmíněn heslem. Zmíněná adaptivita bude spočívat v tom, že bude uživateli umožněn opětovný vstup do privátních her bez opětovné autorizace, případně mu budou již navštívené a používané hry zvýrazněny.

Vstupem do hry dojde ke vstupu do části aplikace, kde budou všechny informace a všechny možnosti a funkce souviset pouze s touto konkrétní instancí, která nebude nijak ovlivněna ani nebude ovlivňovat ostatní. Funkcionalita této sekce může být pomyslně rozdělena na část kde vykonávání funkcí spouští uživatel a na část kde se o spuštění stará systém automaticky.

Jednou z hlavních odlišností oproti popisovaným konkurenčním online aplikacím bude nenucená příslušnost uživatele k výukovým skupinám a fungování simulace i bez dohlížející osoby. Nicméně vytváření a nastavení herních instancí bude stále v režii této osoby, tudíž je nutné zahrnout do požadavků i administrační část, kde bude možné spravovat jak nové tak probíhající hry. Tato část bude opět vztažena pouze na jednu z nich, kde vstup bude podmíněn autorizačními pravidly popsány níže.

5.2 Aktéři systému

Po nasazení je nutné očekávat přístup více uživatelů zároveň. Systém bude schopen rozpoznat jejich identitu a označit je jednou z uživatelských rolí. Pro anonymního uživatele bude přístupná jen vstupní informační část aplikace, a nebude pro něj v systému vytvořena regulérní role, nicméně v dalších částech této kapitoly figurovat bude. Ostatní budou v systému popsáni následovně:

- Uživatel: Tuto roli automaticky obdrží anonymní uživatel po registraci. Role má právo na přístup jak do veřejných herních instancí, tak, v případě korektní autorizace, i do těch privátních, kde může upravovat a řídit vlastní společnosti.
- Administrátor: Uživatel v roli administrátora má právo spravovat instance her a tedy i definovat herní prostředí a zasahovat do samotného průběhu. Navíc má právo správy registrovaných uživatelů a tedy i přidělování uživatelských rolí.

Konkrétnější privilegia jednotlivých rolí budou popsány níže formou případů užití.

5.3 Případy užití

Na základě funkcionálního popisu aplikace ve čtvrté kapitole jsou v následujícím textu jednotlivé funkce popsány jako posloupnost kroků vyjadřující interakci mezi uživatelem a systémem,

označující se jako scénáře případu užití. Ty jsou rozčleněny dle odpovídajících částí popsaných v sekci 5.1.

Kromě textového popisu jsou u každé části formou diagramu případu užití schematicky vyobrazeny vazby mezi aktéry a funkcemi, jež jsou oprávněni vykonávat. Popisovány budou jen scénáře, kde není jejich funkcionality na první pohled zřejmá, nebo mají komplexnější strukturu. Pokud je ve scénáři prováděno validování vstupních hodnot a dopadne-li tato validace neúspěšně, je předpokládáno zamezení další činnosti a uvědomění uživatele o vzniklé anomálii. Tato alternativa tedy nebude do scénářů zahrnována.

5.3.1 Výchozí část

Je zřejmé, že funkcionality této části se bude zaměřovat hlavně na autorizaci a autentizaci uživatelů a řízení jejich přístupu k jednotlivým herním instancím. Odtud je patrné vystupování jak přihlášeného tak anonymního uživatele.



Obrázek 5.1- Diagram případu užití výchozí části

1. Vstup do hry

Aktér: Uživatel

Hlavní scénář:

1. Aktér vybere požadovanou hru
2. Systém zobrazí formulář pro editaci fiktivní postavy podnikatele, s jakou ve hře bude uživatel vystupovat a je-li hra privátní, tak i pole pro zadání vstupního klíče
3. Aktér vyplní požadované údaje a odešle formulář
4. Systém validuje a uloží hodnoty do databáze a nastaví fiktivní postavě počáteční finanční částku dle nastavení hry
5. Systém uživatele přesměruje do zvolené hry

Alternativní scénáře:

- 2a. Uživatel již působí v aktuální instanci
 1. Pokračuje se krokem 5

2. Opuštění hry

Aktér: Uživatel

Prekondice:

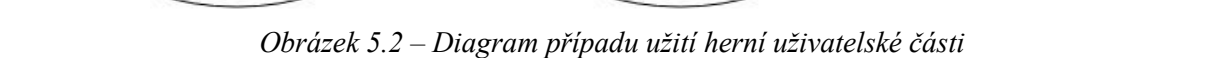
- Aktér působí v dané instanci hry

Hlavní scénář:

1. Aktér zvolí opuštění hry
2. Systém vymaže všechny aktérem vytvořené entity včetně jeho postavy podnikatele v dané hře

5.3.2 Herní uživatelská část

V herní části se bude vždy pracovat pouze s jednou konkrétní instancí hry, a je tedy nutné předem zajistit její zvolení. Pro následující scénáře bude předpokládáno, že tuto podmínku splňují a nebude u nich uváděna v sekci prekondice.



Akteři: Uživatel

331 332 333 334

1000

2. Systém zobrazí seznam regionů aktuální herní instance v přehledném seznamu
3. Aktér vybere region z interaktivní mapy s možností posunu v mapě, nebo z nabízeného seznamu

4. Výběr parcely

Aktéři: Uživatel

Prekondice:

- Musí být zvolen region

Hlavní scénář:

1. Systém zobrazí obrázky budov, dle aktuálního regionu, v interaktivní mapě s izometrickým uspořádáním
2. Aktér vybere parcelu z interaktivní mapy s možností posunu a přibližování

5. Správa nemovitostí

Aktéři: Uživatel

Prekondice:

- Musí být zvolena parcela s veřejnou budovou

Hlavní scénář:

1. Systém zobrazí seznam zakoupených nemovitostí ve veřejné budově a jejich vlastníků
2. Aktér zvolí možnost zakoupení nemovitosti nebo úpravu stávající nemovitosti
3. Systém zobrazí editační formulář a celkovou nákupní cenu za nemovitost navýšenou o cenu standardního vybavení nemovitosti
4. Aktér vyplní a odešle formulář
5. Systém zkontroluje výši aktivního rozpočtu aktéra
6. Systém validuje a uloží hodnoty do databáze
7. Systém zkopíruje standardní vybavení budovy do vybavení nemovitosti
8. Systém vloží jednorázovou platbu za nemovitost a standardní vybavení, a upraví výši aktivního rozpočtu

Alternativní scénáře:

- 2a. Aktér zvolí možnost prodeje nemovitosti
 1. Systém vymaže záznam z databáze
 2. Systém vloží jednorázový zisk za nemovitost a standardní vybavení a upraví výši aktivního rozpočtu
 3. Scénář je ukončen
- 2a. Aktér zvolí úpravu stávající nemovitosti
 1. Systém zobrazí editační formulář
 2. Aktér vyplní a odešle formulář
 3. Systém validuje a uloží hodnoty do databáze a ukončí scénář

5a. Aktér nemá dostatečné finanční prostředky

1. Systém zobrazí chybovou zprávu a ukončí scénář

6. Správa společností

Aktéři: Uživatel

Prekondice:

- Musí být zvolena parcela s budovou radnice

Hlavní scénář:

1. Systém zobrazí seznam společností dle aktéra
2. Aktér zvolí přidání nové společnosti, nebo úpravu stávající
3. Systém zobrazí editační formulář, jehož struktura odpovídá zvolenému typu společnosti
4. Aktér vyplní a odešle formulář
5. Pokud se jedná o nový záznam, systém zkontroluje výši aktivního rozpočtu aktéra, vloží jednorázovou platbu za vytvoření živnostenského oprávnění a upraví výši tohoto rozpočtu
6. Systém validuje a uloží hodnoty do databáze
7. Systém nastaví živnostenské oprávnění jako aktuálně vybranou společnost

Alternativní scénáře:

- 2a. Aktér zvolí smazání živnostenského oprávnění
 1. Systém smaže záznam z databáze a ukončí scénář
- 5a. Pokud se jedná o nový záznam a aktér nemá dostatečné finanční prostředky
 1. Systém zobrazí chybovou zprávu, a ukončí scénář

7. Správa zaměstnanců

Aktéři: Uživatel

Prekondice:

- Musí být zvolena parcela s budovou úřadu práce

Hlavní scénář:

1. Aktér zvolí možnost zobrazení uchazečů o práci
2. Systém zobrazí seznam uchazečů o práci v daném regionu
3. Aktér zvolí možnost najmutí žadatele, přičemž si může prohlédnout jeho životopis, a tak zjistí míru jeho schopností
4. Systém zobrazí editační formulář
5. Aktér vyplní a odešle formulář
6. Systém validuje a uloží hodnoty do databáze
7. Systém vloží trvalou platbu za zaměstnance

Alternativní scénáře:

- 1a. Aktér zvolí možnost zobrazení svých zaměstnanců za účelem jejich propuštění

1. Systém zobrazí seznam zaměstnanců v aktivní společnosti
 2. Aktér zvolí možnost propuštění zaměstnance
 3. Systém vymaže záznam a všechny jeho závislosti z databáze a ukončí provádění
- 1b. Aktér zvolí možnost zobrazení svých zaměstnanců za účelem editace jejich pracovních poměrů
1. Systém zobrazí seznam zaměstnanců v aktivní společnosti
 2. Aktér zvolí možnost editace pracovního poměru zaměstnance
 3. Systém zobrazí editační formulář
 4. Aktér vyplní a odešle formulář
 5. Systém validuje a uloží hodnoty do databáze a ukončí scénář

8. Vylepšování zaměstnanců

Aktéři: Uživatel

Prekondice:

- Zvolený zaměstnanec musí mít dostupný nenulový počet bodů pro vylepšení

Hlavní scénář:

1. Systém zobrazí seznam zaměstnanců v aktivní společnosti
2. Aktér zvolí možnost zobrazení životopisu zaměstnance, u kterého chce vylepšit jeho schopnosti
3. Systém zobrazí životopis zaměstnance a tedy i výčet jeho schopností s aktuální výší bodového rozdělení
4. Aktér dle uvážení opakovaně inkrementuje konkrétní schopnosti zvolením přidruženého tlačítka až do vyčerpání dostupných bodů k vylepšení
5. Aktér odešle formulář
6. Systém validuje a uloží hodnoty do databáze

9. Poskytnutí úvěru

Aktéři: Uživatel

Prekondice:

- Musí být zvolena parcela s budovou banky

Hlavní scénář:

1. Systém zobrazí seznam všech poskytnutých úvěrů dle vybrané společnosti
2. Aktér zvolí možnost poskytnutí nového úvěru
3. Aktér vyplní a odešle editační formulář
4. Systém validuje hodnoty, provede výpočet výše splátky a uloží je do databáze jako trvalou platbu
5. Systém vloží jednorázový zisk ve výši požadované půjčky
6. Systém upraví výši aktivního rozpočtu

10. Výběr vybavení

Aktéři: Uživatel

Prekondice:

- Musí být zvolena nemovitost

Hlavní scénář:

1. Systém zobrazí obrázky vybavení, dle aktuální nemovitosti, v interaktivní mapě s izometrickým uspořádáním, kde pokud zvolená nemovitost nepatří aktuálně vybrané společnosti, omezí se zobrazené vybavení jen na prodejní regály a pokladny
2. Aktér vybere vybavení z interaktivní mapy s možností posunu a přibližování

11. Správa umístění vybavení

Aktéři: Uživatel

Prekondice:

- Musí být zvolena nemovitost patřící aktuálně vybrané společnosti

Hlavní scénář:

1. Provede se scénář 10 bez nutnosti výběru konkrétního vybavení
2. Systém zobrazí seznam dostupných typů vybavení
3. Aktér zvolí typ vybavení, které chce nově umístit
4. Systém umístí nové vybavení do aktuálního středu mapy a označí ho jako editované
5. Aktér může zvolením pozice v mapě určit pozici vybavení, případně může vybavení natočit
6. Systém zkontroluje, zda je pozice vybavení validní a eventuálně povolí možnost uložení
7. Aktér uloží změny
8. Systém zkontroluje výši aktuálně vybraného rozpočtu, vloží jednorázovou platbu za vybavení a upraví tento rozpočet
9. Systém validuje a uloží hodnoty do databáze

Alternativní scénáře:

- 3a. Aktér použije uskutečněný výběr vybavení ve scénáři 10 a zvolí jeho přesun
 - a. Scénář pokračuje krokem 5
- 3b. Aktér použije uskutečněný výběr vybavení ve scénáři 10 a zvolí jeho prodej
 - a. Systém vymaže záznam z databáze
 - b. Scénář je ukončen
1. Aktér nemá dostatečné finanční prostředky
 - a. Systém zobrazí chybovou zprávu, a ukončí scénář

12. Výběr produktu

Aktéři: Uživatel

Hlavní scénář:

1. Aktér žádá zobrazení produktu, které jsou k dispozici pro danou společnost
2. Systém zobrazí stránkovaný seznam všech dostupných produktů, u kterých bylo pro danou společnost provedeno vytvoření prototypu
3. Aktér může produkty filtrovat zvolením položky ze stromu kategorií produktu
4. Aktér vybere požadovaný produkt
5. Systém zobrazí detail produktu
6. Aktér potvrdí výběr produktu

13. Správa aktivit produktu

Poznámka:

- Tento scénář je generický a sjednocuje funkčnost scénářů *správa sortimentu*, *správa skladování produktu*, *správa výroby produktu*, *správa transportů produktu* a *správa designů produktu*

Aktér: Uživatel

Prekondice:

- Musí být vybrán odpovídající typ vybavení pro konkrétní scénář (*správa sortimentu* = prodejní regál, *správa skladování produktu* = skladovací regál, *správa výroby produktu* = konstrukční stůl, *správa transportů produktu* = automobil, *správa designů produktu* = návrhová tabule)
- Musí být zvolena nemovitost patřící aktuálně vybrané společnosti

Hlavní scénář:

1. Systém zobrazí seznam entit dle vybraného vybavení
2. Aktér zvolí přidání nové entity, pokud nebylo dosaženo jejich maximálního počtu, nebo vybere úpravu stávající entity, pokud jde o scénář *správa sortimentu*
3. Systém zobrazí editační formulář
4. Provede se scénář 12
5. Systém, pokud je to třeba, vypočítá nezbytné údaje v závislosti na uživatelských vstupech (například celkovou cenu na výrobu)
6. Aktér vyplní a odešle editační formulář
7. Systém zkontroluje výši aktuálně vybraného rozpočtu, vloží jednorázovou platbu a upraví tento rozpočet (pouze scénáře *správa výroby produktu* a *správa designů produktu*)
8. Systém validuje a uloží hodnoty do databáze

Alternativní scénáře:

- 2a. Aktér zvolí možnost smazání entity
 1. Systém smaže záznam z databáze a ukončí scénář
- 7a. Aktér nemá dostatečné finanční prostředky
 1. Systém zobrazí chybovou zprávu, a ukončí scénář

14. Přiřazení zaměstnanců

Aktéři: Uživatel

Prekondice:

- Musí být vybráno vybavení, ke kterému je možné přiřadit zaměstnance
- Musí být zvolena nemovitost patřící aktuálně vybrané společnosti

Hlavní scénář:

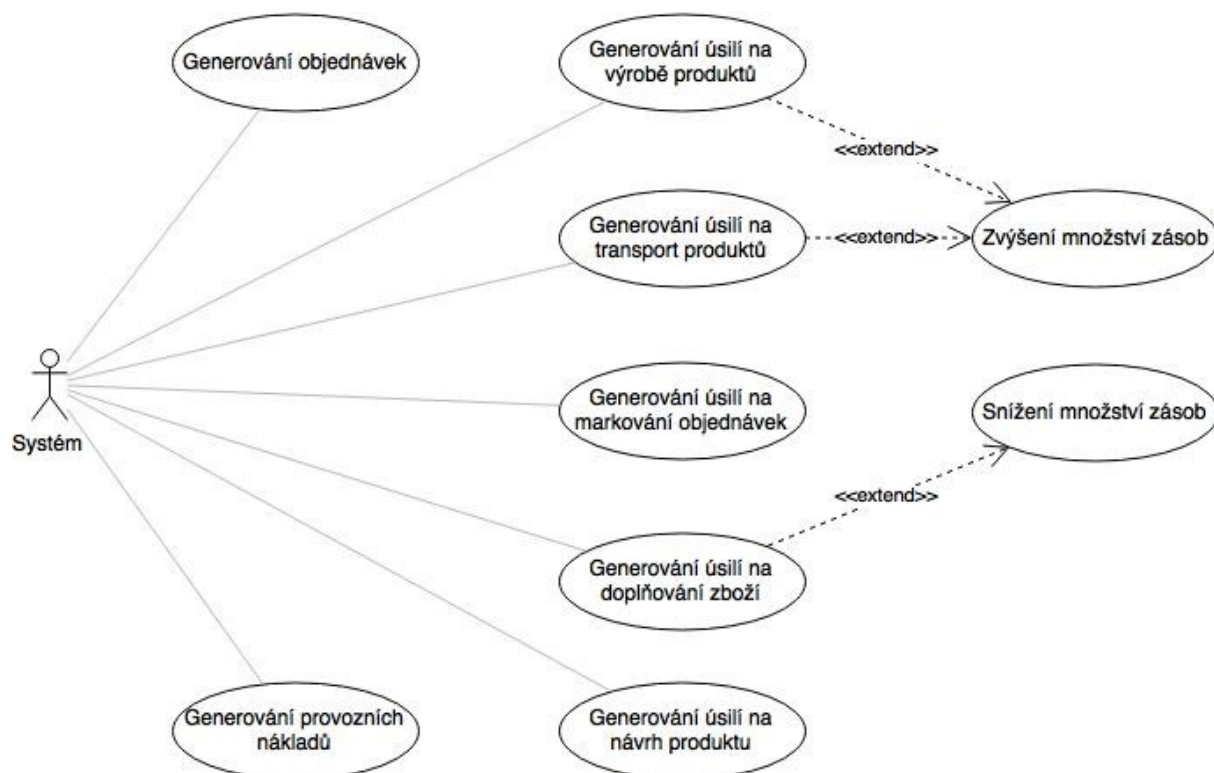
1. Systém zobrazí přiřazené zaměstnance k danému vybavení
2. Aktér zvolí přidání nového záznamu, pokud nebylo dosaženo jejich maximálního počtu, nebo úpravu stávajícího
3. Systém nabídne aktérovi seznam zaměstnanců, kteří pracují v aktuálně zvolené nemovitosti
4. Aktér vyplní a odešle editační formulář (zaměstnanec, podíl práce)
5. Systém provede součet podílu práce pro všechna vybavení, kde je tento konkrétní zaměstnanec přiřazen a je-li to třeba, tak sníží podíl práce ostatních entit, tak aby součet nepřesáhl hodnotu 100

Alternativní scénáře:

- 2a. Aktér zvolí smazání stávající záznamu
 1. Systém vymaže záznam z databáze a ukončí scénář

5.3.3 Herní systémová část

Scénáře v této části budou systémem prováděny automaticky bez možnosti jakékoliv uživatelské interakce. Detailní popis jejich funkcionality je v kontextu této kapitoly irelevantní a bude mu věnován dostatečný prostor až v těch následujících.



Obrázek 5.3 - Diagram případu užití herní systémové části

5.3.4 Administrační část

Stejně jako v herní uživatelské části popisované v kapitole 5.3.2 i zde je předpokládáno prvotní zvolení konfigurované hry, proto je předpokládána obdobná prekondice.



Obrázek 5.4 - Diagram případu užití administrační části

15. Výběr obrázku

Aktéři: Administrátor

Hlavní scénář:

1. Systém zobrazí stránkovaný seznam obrázků z určité složky
2. Aktér zvolí možnost nahrání obrázku z počítače
3. Systém načte obrázek, zkontroluje datový typ a eventuálně vytvoří miniatury v předdefinovaných velikostech
4. Systém aktualizuje seznam obrázků
5. Aktér vybere požadovaný obrázek a potvrdí volbu

Alternativní scénáře:

- 2a. Požadovaný obrázek je již nahraný
 1. Scénář pokračuje krokem 5

6 Návrh aplikace

Před samotnou implementací aplikace je třeba vyhotovit její návrh, který je předmětem právě této kapitoly. Zde nebudou brány v úvahu konkrétní vývojové možnosti ani technologie, nýbrž bude popisována logická a datová struktura aplikace se snahou o začlenění doporučených postupů, jako jsou návrhové vzory.

6.1 Doménový model

Struktura doménového modelu navrhované aplikace je poměrně komplexní, a tak je rozdělena do několika částí, kdy se každá věnuje určité funkcionalitě. Patří zde výchozí část (kapitola 6.1.1) a dále pak části, které pracují s konkrétní hrou. To zahrnuje části zodpovídající za vyobrazení jak map exteriérů (kapitola 6.1.3), tak map interiérů budov (kapitola 6.1.4), část řešící problematiku společnosti (6.1.5), část zastřešující práci se zbožím od definice produktů až po logistiku společnosti (kapitola 6.1.6), část pro práci s lidmi, kde spadají jak zákazníci, tak i zaměstnanci a podnikatelé (kapitola 6.1.7), a v poslední řadě, část pro manipulaci s trhem (kapitola 6.1.8).

Vyjma výchozí části spolu tyto sekce vzájemně spolupracují, jsou propojeny a do jisté míry i na sobě závislé. Mají také přístup do společné sekce (kapitola 6.1.2), která jednak zabezpečuje rutinní funkcionalitu aplikace a jednak se snaží o abstrakci zmíněných částí. Výchozí část je od ostatních logicky oddělena, aby bylo možno dosáhnout nezávislosti na hardwarových prostředcích.

V následujících kapitolách budou části blíže specifikovány, včetně popisu jednotlivých tříd a jejich vazeb. U náležitých UML třídních diagramů jsou **modrou** barvou vyznačeny třídy, které logicky spadají do odlišné sekce, ovšem jsou uvedeny pro znázornění vazby. Třídy jsou zde uvedeny bez vlastností a metod pro zachování přehlednosti, ty budou popsány lineárním textovým zápisem. Lineární zápis obsahuje všechny vlastnosti entit primitivního datového typu včetně znázornění primárních, *cizích* klíčů a „odvozených vlastností“ ovšem bez těch, které byly převzaty z předka.

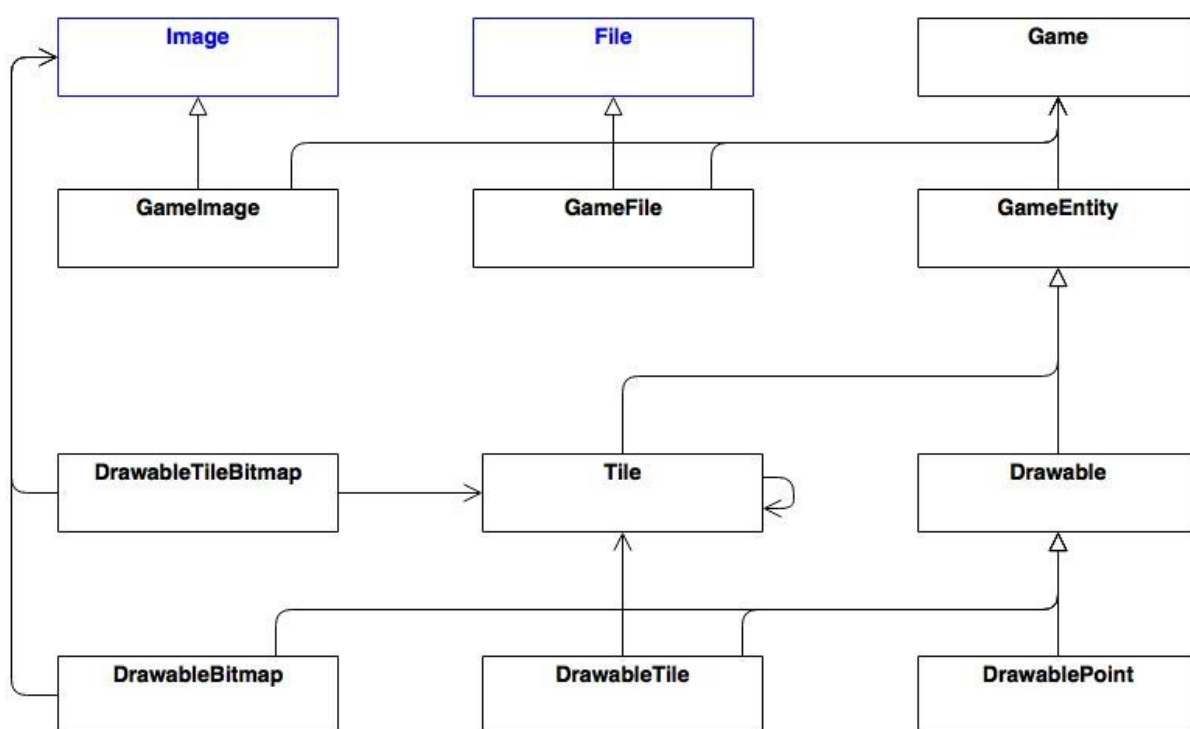
6.1.1 Výchozí část

Doménový model výchozí části je používán výhradně v případech užití stejnojmenné sekci (viz 5.3.1). Obsahuje pouze třídy `Game` a `User`.

`Game` (`Id`, `ConnectionString`, `RegisterKey`, `Name`, `MoneyInit`, „`IsLocked`“)

`User` (`Email`, `Password`, `Role`, `LoginDateTime`, `RegisterDateTime`)

Třída `Game` reprezentuje instanci hry. Ke každé se vztahují určité konfigurační vlastnosti jako `Name`, nesoucí název hry, `RegisterKey` určující přístupový klíč, který umožňuje nastavit hru jako privátní (odvozená binární vlastnost `IsLocked` bude v tomto případě nastavena na `true`), nebo `MoneyInit`, která určuje počáteční výši uživatelského rozpočtu. Jelikož mohou být data každé hry v různých databázích, je nutné uchovávat přístupové údaje k nim (vlastnost `ConnectionString`). Třída `User` nese informace o uživateli, to znamená jeho email (vlastnost `Email`), heslo (vlastnost



Ve výchozí části doménového modelu (kapitola 6.1.1) existuje stejně jako v této části třída `Game`. Zde je tato třída pouhým nezávislým prostředníkem se stejným způsobem identifikace (vlastnost `Id`). Na ni jsou pak závislé třídy `GameEntity`, `GameFile` a `GameImage`, jež reprezentují základní entitu, soubor a obrázek náležící k určité herní instanci. Třída `Drawable` reprezentuje objekt, který je nějakým způsobem graficky zobrazitelný na mapě. To vyžaduje informaci o poloze (vlastnosti `X` a `Y`) objektu a jeho pořadí pro překrytí (vlastnost `ZIndex`). O způsobu zobrazení rozhodují potomci třídy `Drawable`. Potomek `DrawablePoint` zobrazuje bod, jež je symbolizován kruhem určitého poloměru (vlastnost `Size`) a určité barvy (vlastnost `Color`). Bod je vykreslen i s popiskem definovaným vlastností `Name`. Potomek `DrawableBitmap` zobrazuje bitmapu s přidruženým obrázkem. Poslední potomek `DrawableTile` zobrazuje izometrický objekt (třída `Tile`) pomocí jednoho ze čtyř možných izometrických obrázků (třída `DrawableTileBitmap`) s ohledem na natočení (vlastnost `RotationIndex`). Objekt třídy `Tile` má definovány své rozměry jak na ose `x` vlastností `Width`, tak na ose `y` vlastností `Height`. Odvozené vlastnosti `Width` a `Height` třídy `DrawableTile` pak s těmito rozměry pracují, ovšem s ohledem na natočení objektu. Mimo operaci otáčení je také možno objekt vykreslovat opakovaně v příslušných osách. K tomu slouží vlastnosti `ScaleX` a `ScaleY`, jež určují počet opakování, přičemž jejich hodnota musí být od 1 až do hodnoty vlastností `ScaleXMax` a `ScaleYMax` třídy `Tile`. Dosud nebyly zmíněny omezení při umísťování objektu. Ten samozřejmě nemůže být umístěn tak, aby překrýval jiný objekt, ovšem může být umístěn na objekt s menším pořadím pro překrytí. Pokud je třeba blíže specifikovat, o jaký objekt se má jednat, je možné použít nepovinnou rekurzivní vazbu třídy `Tile`.

6.1.3 Exteriér

Primární účel této části modelu je korektní znázornění jak mapy prostředí, tak i mapy regionu. O model vykreslení se z velké části stará abstrakce popsaná v předchozí kapitole. Zde jsou popsány konkrétní prvky, které lze konkrétním způsobem vykreslit a jejich návaznost na tuto abstrakci (viz Obrázek 6.2).

Obrázek 6.2 – Doménový model – exteriér

Building (Name)

BuildingBank (InterestRate, LoanMax)

BuildingImageRotation (BuildingId, BuildingGameId, BuildingImageId, BuildingImageGameId)

BuildingPublic (PriceBuy, Attendance, „FKBuildingEquipmentsPriceBuy“)

BuildingLabourOffice (InsuranceHealth, InsuranceSocial)

BuildingTownhall (PriceCompanyCreate)

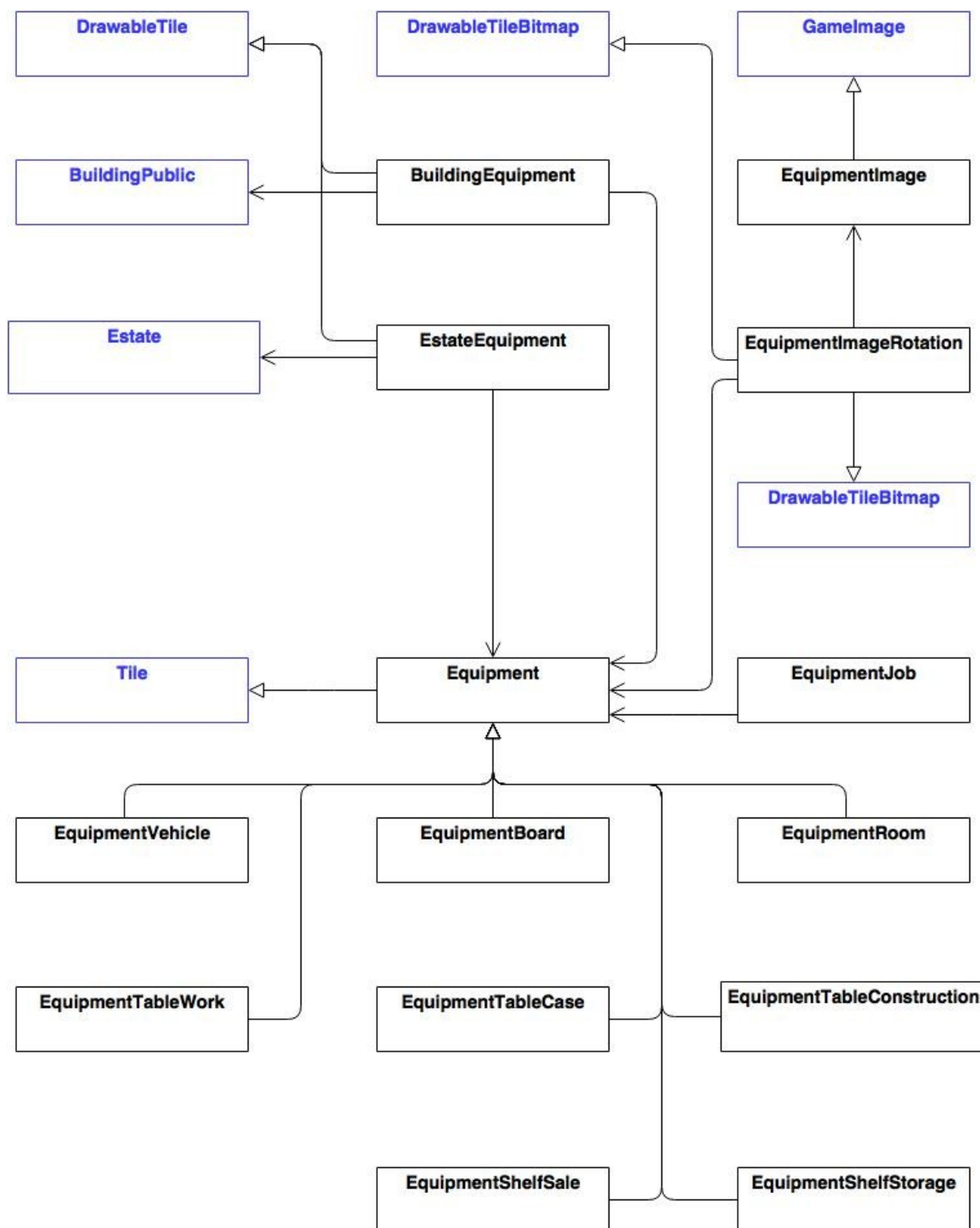
MapLayer (Name, *MapLayerImageId*, *MapLayerImageGameId*)Parcel (*RegionId*, *RegionGameId*, *BuildingId* *BuildingGameId*)

Region (Name, CenterX, CenterY)

Pro mapu prostředí jsou nejdůležitější třídy `MapLayer` a `Region`, které reprezentují graficky znázornitelné objekty, což dokládá jejich odvození ze tříd `DrawableBitmap` a `DrawablePoint`. Třída `MapLayer` reprezentující podklad mapy prostředí je tedy vykreslena jako obrázek, jehož zdroj nese třída `MapLayerImage`. Pro snadnější orientaci v práci s vrstvami je možné si dané položky pojmenovat pomocí vlastnosti `Name`. Třída `Region` je vykreslena jako bod na těchto vrstvách. Rovněž je možné region pojmenovat a mimo to i nastavit souřadnice středu náležící mapy regionu (vlastnosti `CenterX` a `CenterY`). Ty jsou použity pro určení vzdálenosti mezi regiony, což je údaj potřebný při transportu zboží. Pro mapu regionu je pak důležitá třída `Building`, která reprezentuje konkrétní typ budovy nebo struktury, a třída `Parcel` nesoucí informace o konkrétním umístění této budovy nebo struktury. Ty jsou hierarchicky členěny dle typů na veřejné (`BuildingPublic`), dekorativní (`BuildingDecoration`) a oficiální (`BuildingOfficial`), které jsou dále konkretizovány na banku (`BuildingBank`), úřad práce (`BuildingLabourOffice`) a radnici (`BuildingTownhall`). Některé typy mají specifické vlastnosti. U veřejných budov je nutné znát procentuální vyjádření návštěvnosti zákazníků (vlastnost `Attendance`) a cenu na nákup nemovitosti, která se skládá z ceny samotné nemovitosti (vlastnost `PriceBuy`) a s ceny standardního vybavení (vlastnost `FKBuildingEquipmentsPriceBuy`). Pro banku je důležitým údajem úroková míra (vlastnost `InterestRate`) a maximální výše půjčky (vlastnost `LoanMax`). Pro pracovní úřad je důležitá procentuální výše zdravotního a sociálního pojištění (vlastnosti `InsuranceHealth`, `InsuranceSocial`). A pro radnici je to výše poplatku při založení společnosti (vlastnost `PriceCompanyCreate`).

6.1.4 Interiér

Další část doménového modelu se věnuje vykreslování mapy pobočky. Struktura je obdobná jako struktura použitá pro vykreslování mapy regionu a je také použita stejná abstrakce ze sdílené části (viz Obrázek 6.3).



Obrázek 6.3 – Doménový model – interiér

BuildingEquipment (*EquipmentId*, *EquipmentGameId*, *BuildingPublicId*, *BuildingPublicGameId*)

Equipment (Name, PriceBuy, WorksCountLimit)

EquipmentBoard (DesignsCountLimit)

EquipmentImageRotation (*EquipmentId*, *EquipmentGameId*, *EquipmentImageId*, *EquipmentImageGameId*)

EquipmentJob (*EquipmentId*, *EquipmentGameId*, *Job*, StepEffortMin, StepEffortMax, EffortXp, „JobString“)

EquipmentShelfSale (VolumeCapacity, SortimentsCountLimit)

EquipmentShelfStorage (VolumeCapacity, StocksCountLimit)

EquipmentTableCase (OrdersCountLimit)

EquipmentTableConstruction (ConstructionsCountLimit)

EquipmentVehicle (VolumeCapacity, TransportsCountLimit)

EstateEquipment (*EquipmentId*, *EquipmentGameId*, *EstateId*, *EstateGameId*)

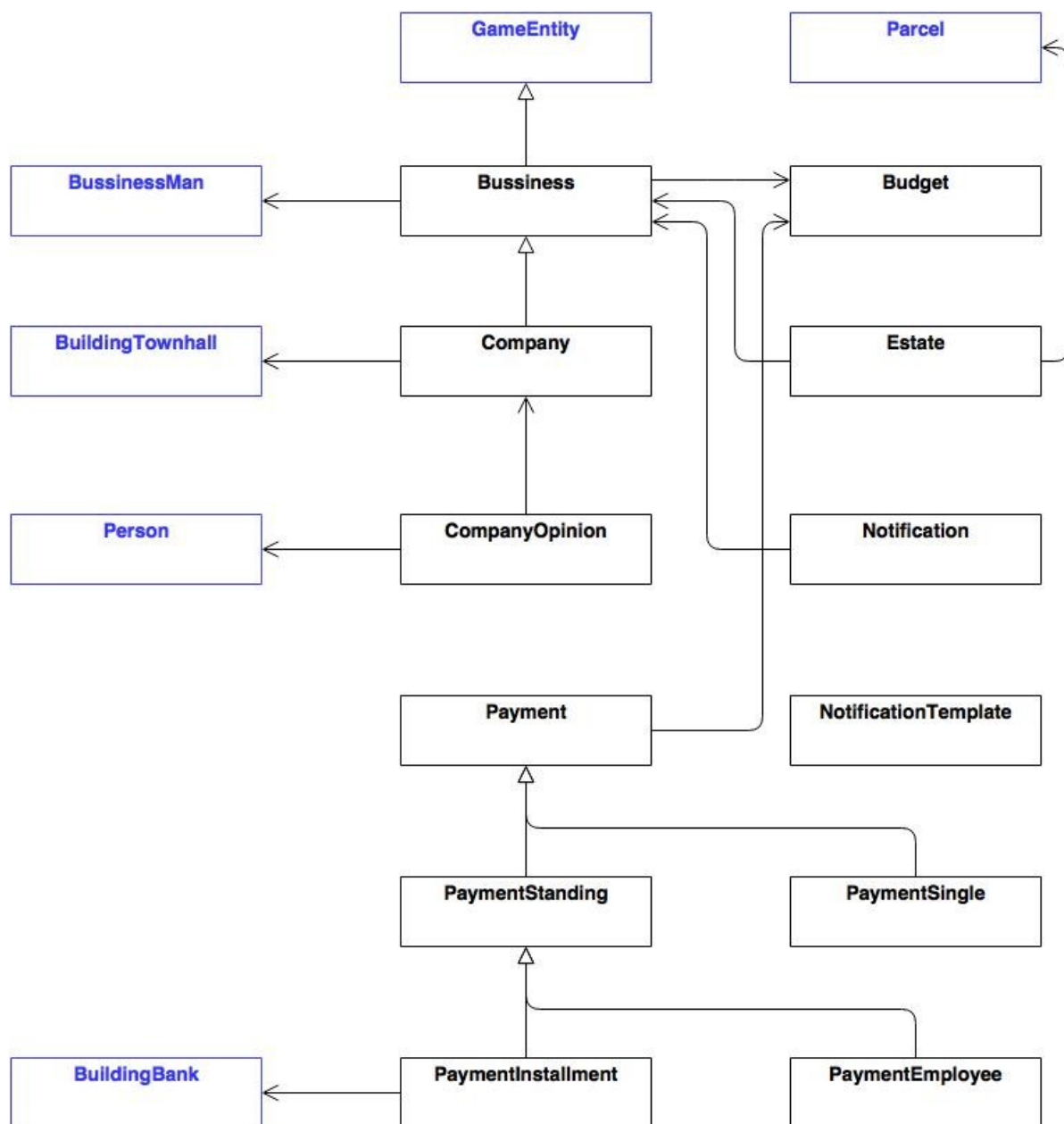
Vybavení nemovitosti je reprezentováno potomky třídy Equipment, konkrétně třídou EquipmentBoard pro návrhovou tabuli, EquipmentShelfSale pro prodejní regál, EquipmentShelfStorage pro skladovací regál, EquipmentTableCase pro pokladnu, EquipmentTableConstruction pro konstrukční stůl, EquipmentTableWork pro pracovní stůl a EquipmentVehicle pro transportní vozidlo. Posledním potomkem třídy Equipment je třída EquipmentRoom, která zastupuje místnosti v nemovitosti. Třída Equipment nese společné informace o názvu vybavení (vlastnost Name), ceně vybavení (vlastnost PriceBuy) a limitu přiřazených pracovníků (vlastnost WorksCountLimit). Většina potomků pak má definován limit počtu náležitých, aktuálně probíhajících pracovních činností (vlastnosti DesignsCountLimit, SortimentsCountLimit, StocksCountLimit, OrdersCountLimit, ConstructionsCountLimit, TransportsCountLimit). Prodejní, skladovací regály a transportní vozidla mimo to limitují i počet přidružených produktů, respektive limitují jejich celkový objem, a to pomocí vlastnosti VolumeCapacity. Ke všem zmíněným třídám reprezentující vybavení, se váže třída EquipmentJob, jež nese informace o dostupných pracovních činnostech. Vlastnosti StepEffortMin a StepEffortMax definují rozsah, v jakém se může pohybovat míra vynaloženého úsilí v jednom tahu s ohledem na schopnosti zaměstnance a jeho přiřazený podíl pracovní doby. Vlastnost EffortXp mapuje jednotky vynaloženého úsilí na jednotky zkušeností.

Aby bylo možné uchovávat informace o umístění vybavení v konkrétní nemovitost a také definovat standardní umístění vybavení ve veřejné budově, jsou zde přidány třídy EstateEquipment, s vazbou na nemovitost, a BuildingEquipment, s vazbou na veřejnou budovu.

6.1.5 Společnost

Tato část doménového modelu se zaměřuje na uchovávání informací o společnosti včetně informací o rozpočtu a provedených transakcích, informací o nemovitostech a povědomí veřejnosti.

Mimo to zahrnuje i situace, kdy uživatel žádnou společnost nevlastní a vystupuje pouze jako fyzická osoba.



Obrázek 6.4 – Doménový model – společnost

Budget (Money, Order)

Bussiness (Name, BussinessType, *BussinessManId*, *BussinessManGameId*, *BudgetId*, *BudgetGameId*, „BussinessTypeString“)

Company (*BuildingTownhallId*, *BuildingTownhallGameId*, „Capital“)

CompanyOpinion (*CompanyId*, *CompanyGameId*, *PersonId*, *PersonGameId*, Happiness)

Estate (Name, *BussinessId*, *BussinessGameId*, *ParcelId*, *ParcelGameId*)

Notification (Text, ReadDateTime, *BussinessId*, *BussinessGameId*)

NotificationTemplate (Key, GameId, Text)

Payment (MoneyProfit, PaymentType, *BudgetId*, *BudgetGameId*, „PaymentTypeString“, „MoneyLoss“)

PaymentEmployee (InsuranceHealth, InsuranceSocial, „MoneyLossGross“)

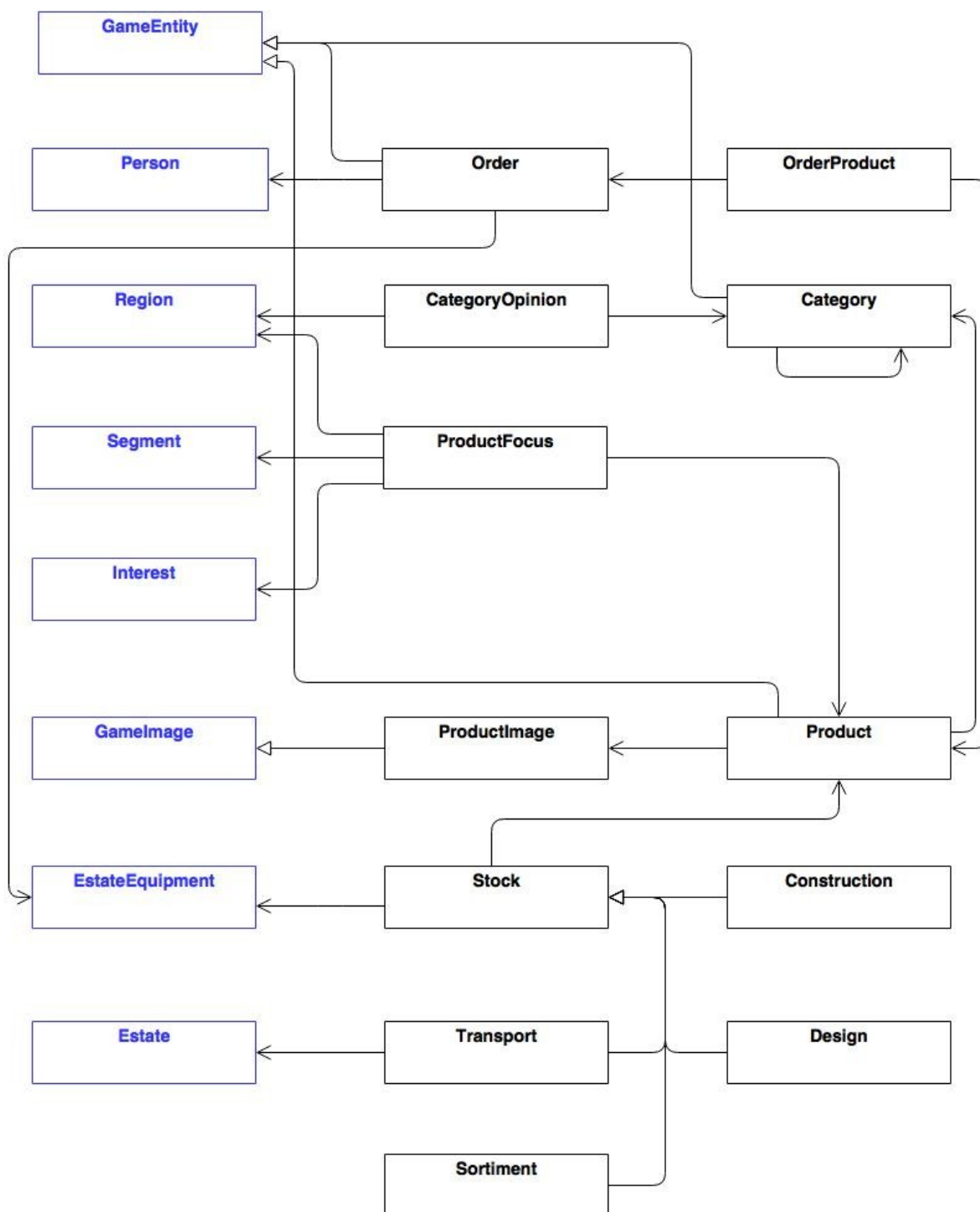
PaymentInstallment (MoneyLoaned, InterestRate, *BuildingBankId*, *BuildingBankGameId*)

PaymentStanding (IterationsTotal, IterationsDone)

Společnosti reprezentuje třída *Company*. Ta je rozšířením třídy *Bussiness*, která zastupuje i fyzickou osobu podnikatele a rozšiřuje ji o vazbu na radnici, kde byla společnost vytvořena, kapitál (vlastnost *Capital*), pokud jde o kapitálový typ společnosti a vazbu na zákazníka, u kterého je ve třídě *CompanyOpinion* uchovávána jeho spokojenost s touto společností (vlastnost *Happiness*). Typ společnosti a jméno společnosti jsou uchovány v předkovi ve vlastnostech *BussinessType* a *Name*. K němu se pak vážou vlastněné nemovitosti (třída *Estate*) a rozpočet (třída *Budget*), jež je používán pro provádění finančních transakcí. Finanční transakcí může být jednorázová platba (třída *PaymentSingle*), nebo trvalá platba (třída *PaymentStanding*), která se pak dále dělí na výplatu zaměstnanci (třída *PaymentEmployee*), s rozšiřujícími vlastnostmi pro hodnotu zdravotního pojištění, sociálního pojištění a hrubé mzdy (vlastnosti *InsuranceHealth*, *InsuranceSocial*, *MoneyLossGross*) a platbu úroku z poskytnutého úvěru (třída *PaymentInstallment*), která nese informace o výši půjčené částky a úrokové míře (vlastnosti *MoneyLoaned*, *InterestRate*). Trvalá platba může mít omezený počet opakování. Proto má třída *PaymentStanding* vlastnosti *IterationsTotal*, pro definici celkového počtu opakování, a *IterationsDone*, pro zaznamenání, kolik opakování již proběhlo. Všechny typy finančních transakcí mají společného předka *Payment*, jež se váže ke konkrétnímu rozpočtu. Jeho aktuální hodnota je ovšem uložena nezávisle ve vlastnosti *Money*. Mimo to obsahuje vlastnost *Order*, která určuje pořadí na trhu dle dosaženého zisku v každém tahu. Platba obsahuje vlastnosti *MoneyProfit* a *MoneyLoss*, které jsou vzájemně opačné a určují finanční profit respektive ztrátu z pohledu uživatele. Dále pak *PaymentType*, specifikující typ platby. Pro notifikaci nějaké nové události v rámci společnosti jsou použity třídy *Notification* a *NotificationTemplate*. Objekt třídy *NotificationTemplate* je pouze šablona notifikační zprávy pro určitý typ události, zatímco objekt třídy *Notification*, je konkrétní notifikace, jejichž text (vlastnost *Text*) byl vytvořen na základě této šablony. Objekt také ve vlastnosti *ReadDateTime* uchovává čas, kdy si uživatel notifikaci zobrazil, aby ho bylo možno upozornit na nové nepřečtené.

6.1.6 Zboží

Část doménového modelu zaměřená na zboží v sobě zastřešuje informace o dostupných produktech, jejich vlastnostech a jejich kategorizaci. Ale také i fáze životního cyklu produktu, kterými v rámci společnosti aktuálně prochází, včetně jejich stavu.



Obrázek 6.5 – Doménový model – zboží

Category (Name)

CategoryOpinion (CategoryId, CategoryGameId, RegionId, RegionGameId, PriceSellingExpected, QualityExpected, „RatioExpected“)

Construction („PricePurchase“, „EffortMax“)

Design (IsDesigning, „EffortMax“)

Order (PersonId, PersonGameId, EstateEquipmentId, EstateEquipmentGameId, Effort, IsCashiering, „EffortMax“, „QuantityTotal“, „PriceSellingTotal“)

OrderProduct (OrderId, OrderGameId, ProductId, ProductGameId, Quantity, PriceSelling, AttendanceFactor, DemandFactor, HappinessFactor, PriceSellingFactor, TrendFactor, „TotalFactor“)

Product (Name, Description, Quality, Durability, Volume, PricePurchase, *CategoryId*, *CategoryGameId*, *ProductImageId*, *ProductImageGameId*)

ProductFocus (ProductId, ProductGameId, SegmentId, SegmentGameId, InterestId, InterestGameId, Focus)

Sortiment (PriceSelling, „EffortMax“)

Stock (*ProductId*, *ProductGameId*, *EstateEquipmentId*, *EstateEquipmentGameId*, Quantity, Effort, „QuantityCapacity“)

Transport (*EstateId*, *EstateGameId*, Distance, „EffortMax“)

Produkt (třída *Product*) má mimo jiné vlastnosti *Quality*, pro určení míry jeho kvality, *Durability*, reprezentující jeho trvanlivost a *Volume*, jež udává jeho objem. Také má určeno, na jaké segmenty a zájmy je cílen díky vlastnosti *Focus* ve třídě *ProductFocus*. Tyto vlastnosti jsou důležité jak pro provádění operací v rámci životního cyklu produktu, tak pro generování objednávek.

Produkt musí být zařazen do nějaké kategorie (třída *Category*), jejichž hierarchie je uložena pomocí rekurzivní vazby. Zákazníci v daném regionu mají představu o tom, kolik by měl produkt z dané kategorie stát (vlastnost *PriceSellingExpected*) a jakou by měl mít kvalitu (vlastnost *QualityExpected*), což jsou vlastnosti třídy *CategoryOpinion*.

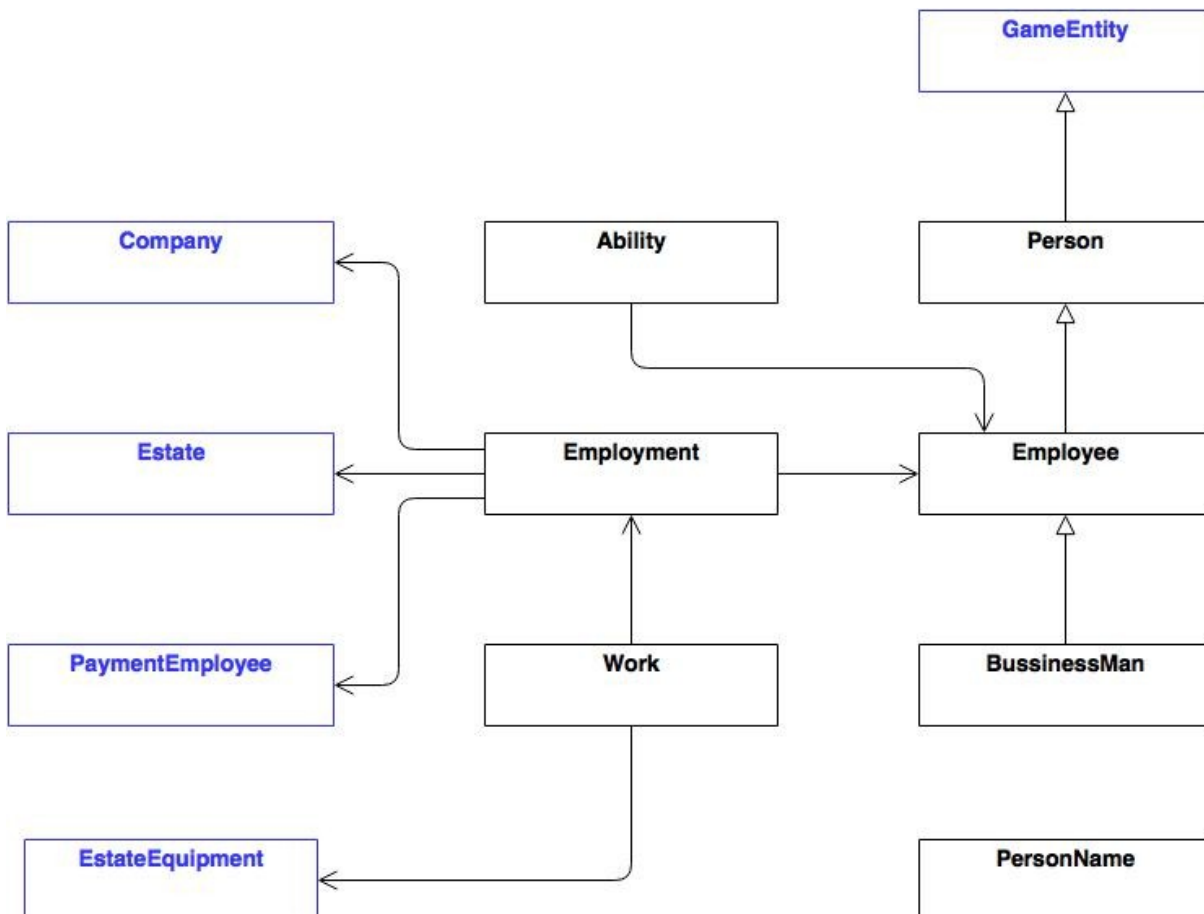
Pro zaznamenání operací s produktem v rámci společnosti je důležitá třída *Stock*. Ta má vazbu jednak na produkt a jednak na konkrétní vybavení umístěné v nemovitosti. Její vlastnost *Quantity* udává aktuální naskladnění, tedy počet kusů produktu, zatímco odvozená vlastnost *QuantityMax* udává maximální možný počet kusů produktu. Vlastnost *Effort* udává aktuální vynaložené úsilí na danou aktivitu, která je reprezentována až konkrétními potomky této třídy. Každý z nich obsahuje odvozenou vlastnost *EffortMax*, která udává množství úsilí nutné k dokončení dané aktivity. Pro výrobu produktu je to potomek *Construction*, pro návrh produktu je to *Design*, pro doplnění zboží k prodeji a zároveň definici prodejního sortimentu včetně stanovení prodejní ceny (vlastnost *PriceSelling*) je to potomek *Sortiment* a pro transport produktu je to potomek *Transport*, který navíc obsahuje vazbu na cílovou nemovitost.

Poslední aktivitou je markování objednávky. Ta je reprezentována třídou *Order*, která zároveň reprezentuje i samotnou zákaznickou objednávku. Stejně jako předchozí aktivity i tato disponuje vlastnostmi *Effort* a *EffortMax*. K objednavce jsou pomocí třídy *OrderProduct* přidruženy jednotlivé produkty, které si zákazník objednává, se specifikací počtu kusů (vlastnost *Quantity*), ceny (*PriceSelling*) a hodnot jednotlivých faktorů, jež ho při koupi ovlivnily. Jedná se o faktor návštěvnosti budovy, poptávky, prodejní ceny, spokojenosti a trendů (vlastnosti

AttendanceFactor, DemandFactor, PriceSellingFactor, HappinessFactor, TrendFactor). Zákaznickou výslednou míru touhy po produktu udává vlastnost TotalFactor.

6.1.7 Lidé

Nedílnou součástí doménového modelu je i část zaměřená na lidské zdroje. Nachází se zde třídy reprezentující jak zákazníka, tak zaměstnance, včetně jeho schopností a pracovního zařazení, ale i uživatelskou postavu podnikatele.



Obrázek 6.6 – Doménový model – lidé

Ability (EmployeeId, EmployeeGameId, Job, Practice, „JobString“)

BussinessMan (UserEmail)

Employee (MoneyExpected, Xp, Level, Happiness, „PracticeImprovement“, „XpLevelCurrent“, „XpLevelUp“)

Employment (EmployeeId, EmployeeGameId, CompanyId, CompanyGameId, PaymentEmployeeId, PaymentEmployeeGameId, EstateId, EstateGameId)

Person (FirstName, LastName, Sex, Income, RegionId, RegionGameId, SegmentId, SegmentGameId, „FullName“, „SexString“)

PersonName (Id, Name, Sex, IsLastName)

Work (EstateEquipmentId, EstateEquipmentGameId, EmploymentEmployeeId, EmploymentEmployeeGameId, EmploymentCompanyId, EmploymentCompanyGameId, Job, Focus, „StepEffortMax“, „StepEffort“, „StepXp“, „JobString“)

Zákazník je reprezentován třídou `Person` a je popsán pomocí základních vlastností jako je jméno (vlastnosti `FirstName`, `LastName` a `FullName`) a pohlaví (vlastnost `Sex`). Má také vazbu na region, ve kterém figuruje a na segment, do kterého spadá.

Od třídy `Person` je odvozená třída `Employee` reprezentující zaměstnance. U té je třeba uchovávat množství zaměstnancových zkušeností (vlastnost `Xp`), jeho úroveň (vlastnost `Level`), jeho spokojenost (vlastnost `Happiness`) a jeho předpokládaný příjem (`MoneyExpected`). Pomocné odvozené vlastnosti `XpLevelCurrent` a `XpLevelUp` vypočítají a vrátí množství zkušeností, jež zaměstnanec potřeboval k postupu na současnou úroveň, a které bude potřebovat k postupu na další. Poslední odvozená vlastnost `PracticeImprovement` udává počet bodů, jež jsou k dispozici pro rozdělení mezi zaměstnancovy schopnosti, kdy jejich aktuální hodnota je uložena ve vlastnosti `Practice` třídy `Ability`.

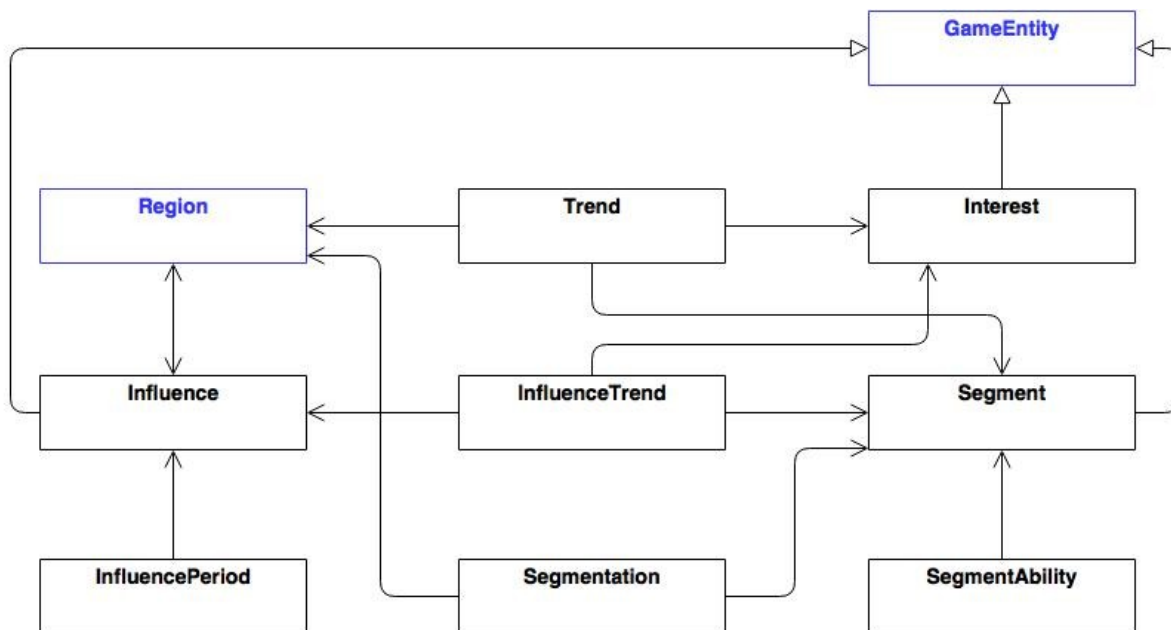
O definici pracovních povinností zaměstnance se stará třída `Employment` v kombinaci s třídou `Work`. Třída `Employment` jednak, díky vazbě na konkrétní společnost (třída `Company`), definuje zaměstnavatele a jednak vazbou na konkrétní nemovitost (třída `Estate`) i místo výkonu práce v rámci společnosti. Je k ní také přiřazena i trvalá platba (třída `PaymentEmployee`) jež představuje zaměstnancův plat. Pomocí třídy `Work` je pak zaměstnanec přiřazen na vykonávání dané pracovní aktivity (vlastnost `Job`) na určitém vybavení (třída `EstateEquipment`) s jistým podílem své pracovní doby, jež je nastavován vlastností `Focus`. Pomocné vlastnosti `StepEffort` a `StepEffortMax` vypočítají množství úsilí, jaké zaměstnanec vyvine za jeden tah, přičemž vlastnost `StepEffort` vypočítá procentuální podíl z vlastnosti `StepEffortMax` ve výši dané hodnotou vlastnosti `Focus`. Poslední vlastností je `StepXp` udávající množství zkušeností, jež zaměstnanec obdrží za jeden tah.

Hierarchii tříd reprezentující osoby zakončuje třída `BusinessMan`, která představuje uživatelskou postavu podnikatele a musí mít tedy vazbu na třídu `User`.

Poslední třídou v této části doménového modelu je třída `PersonName`. Ta nemá žádné vazby a slouží pouze k uložení jak mužských tak ženských (vlastnost `Sex`) jmen a příjmení (vlastnost `Name`), které mají být použity ke generování jmen a příjmení konkrétních instancí tříd `Person` a `Employee`. Hodnota `true` u binární vlastnosti `IsLastName` určuje, že se jedná o příjmení nikoliv křesní jméno.

6.1.8 Trh

Poslední část doménového modelu se zaměřuje na trh a jeho dynamický vývoj. Je zde řešena segmentace trhu, trendy i možnost zavedení událostí, které trh ovlivňují.



Obrázek 6.7 – Doménový model – trh

Influence (Name)

InfluencePeriod (From, To, TimeUnit, InfluenceId, InfluenceGameId)

InfluenceTrend (SegmentId, SegmentGameId, InterestId, InterestGameId, InfluenceId, InfluenceGameId, Focus)

Interest (Name)

Segment (Name, Sex, MoneyFrom, MoneyTo, LevelXp, LevelFactor, HappinessIncreaseFactor, HappinessDecreaseFactor, AttendanceCoef, TrendCoef, DemandCoef, PriceSellingCoef, HappinessCoef, „SexString“)

SegmentAbility (SegmentId, SegmentGameId, Job, PracticeFrom, PracticeTo, „JobString“)

Segmentation (SegmentId, SegmentGameId, RegionId, RegionGameId, Population, Unemployment)

Trend (SegmentId, SegmentGameId, InterestId, InterestGameId, RegionId, RegionGameId, Focus)

Zákazník i zaměstnanec náleží k určitému segmentu (třída Segment), který ovlivňuje jeho chování i vlastnosti. Ten je popsán názvem (vlastnost Name) a pohlavím příslušníků segmentu (vlastnost Sex), jež se použije při generování nových osob. Pro zákazníka jsou dále důležité koeficienty ovlivňující prioritu jednotlivých faktorů, jež zvažuje při nákupu (vlastnosti AttendanceCoef, TrendCoef, DemandCoef, PriceSellingCoef, HappinessCoef) a faktory nárůstu a poklesu jeho nálady (vlastnosti HappinessIncreaseFactor a HappinessDecreaseFactor). Pro zaměstnance je podstatný jeho předpokládaný platový rozsah vymezený vlastnostmi MoneyFrom a MoneyTo, dále základní množství zkušeností pro jednu

úroveň (vlastnost `LevelXp`) a faktor navýšení tohoto množství s přibývajícím úrovní (vlastnost `LevelFactor`).

Pro nově generované zaměstnance je možno pomocí třídy `SegmentAbility` a jejich vlastností `PracticeFrom` a `PracticeTo` vymezit rozsah v jakém se bude pohybovat náhodně vygenerovaná hodnota míry schopnosti dané pracovní aktivity (vlastnost `Job`). Toto je možné definovat různě pro různé segmenty.

Segmenty jsou společně se zájmy, jež jsou reprezentovány třídou `Interest`, přidruženy k regionu pomocí třídy `Trend`, která ve vlastnosti `Focus` uchovává aktuální popularitu. Mimo to je segment k regionu přidružen i třídou `Segmentation` jež udává počet obyvatel daného segmentu v regionu (vlastnost `Population`) a nezaměstnanost daného segmentu v regionu (vlastnost `Unemployment`).

Třída `Influence` představuje událost, ovlivňující trh. Ta má určený časový interval, po který je na trh aplikována (třída `InfluencePeriod`) a samozřejmě i matici hodnot, kterými bude pozměněna popularita jednotlivých segmentů a jejich zájmů (třída `InfluenceTrend`).

6.2 Procedurální náhled

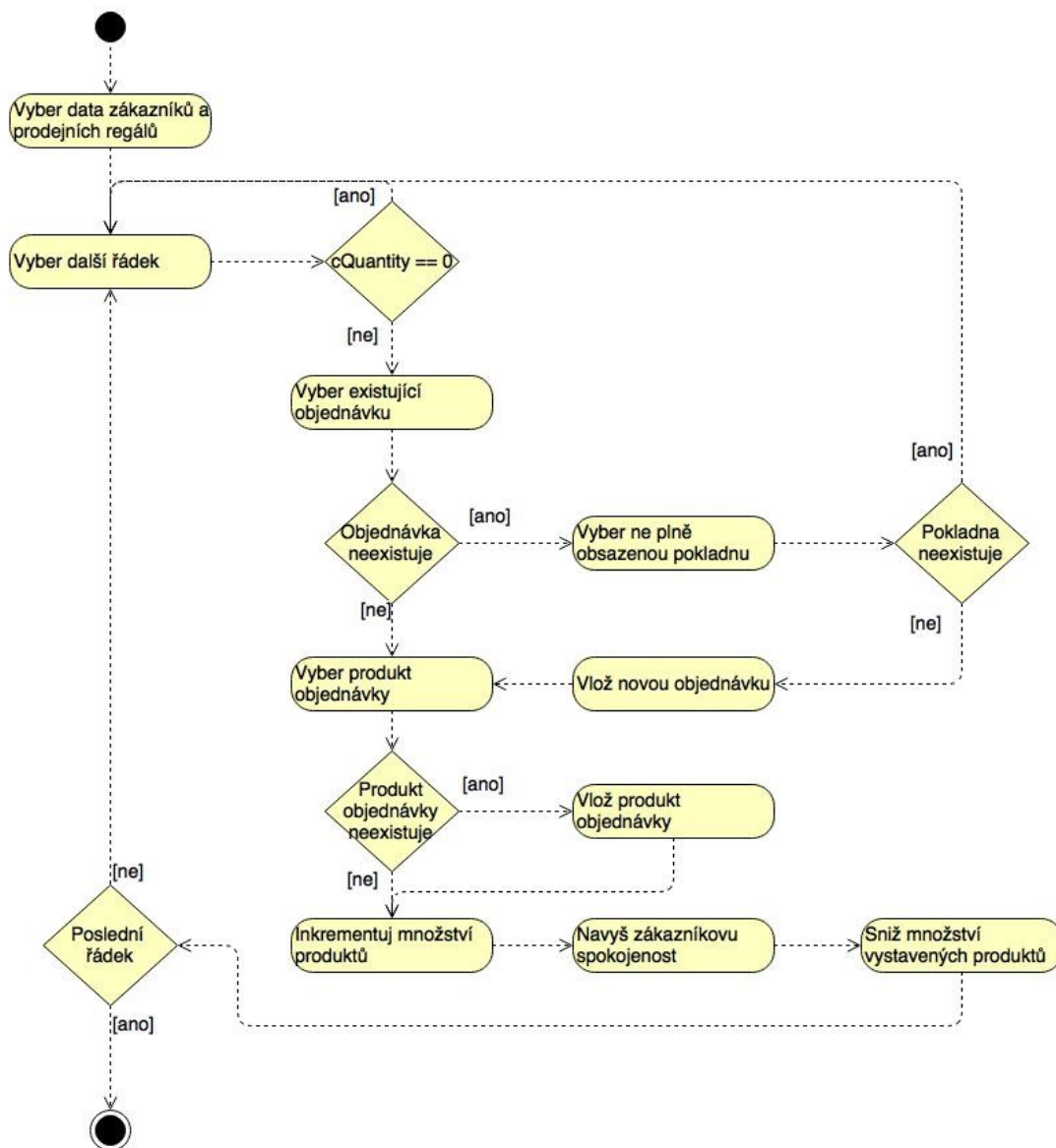
Tato sekce se bude výhradně zabývat detailnějším popisem některých komplexnějších procedur herní systémové části, která byla nastíněna v kapitole 5.3.3.

6.2.1 Generování objednávek

Generování objednávek je jedním z nejdůležitějších funkčních bloků v navrhované aplikaci. Cílem každého uživatele by mělo být právě dosažení co největšího počtu objednávek.

Procedura pro každého zákazníka v náležících regionech, v kombinaci se všemi nabízenými produkty v prodejních regálech všech společností určí, zda má zákazník o produkt zájem.

Je-li produkt vystaven vícekrát, to znamená ve více prodejních regálech, může být zakoupen více než jeden kus. Vybrané kusy pak procedura sváže s vytvořenou objednávkou, kterou přidruží k určité pokladně, přičemž vybírá vždy pokladnu s nejmenším počtem aktuálně markovaných objednávek. V poslední řadě se procedura postará o dekrementaci množství v příslušných prodejních regálech a o navýšení spokojenosti zákazníka dle nastavení v příslušném segmentu. Názornější pohled na proceduru poskytuje Obrázek 6.8.



Obrázek 6.8 – Diagram aktivit pro generování objednávek

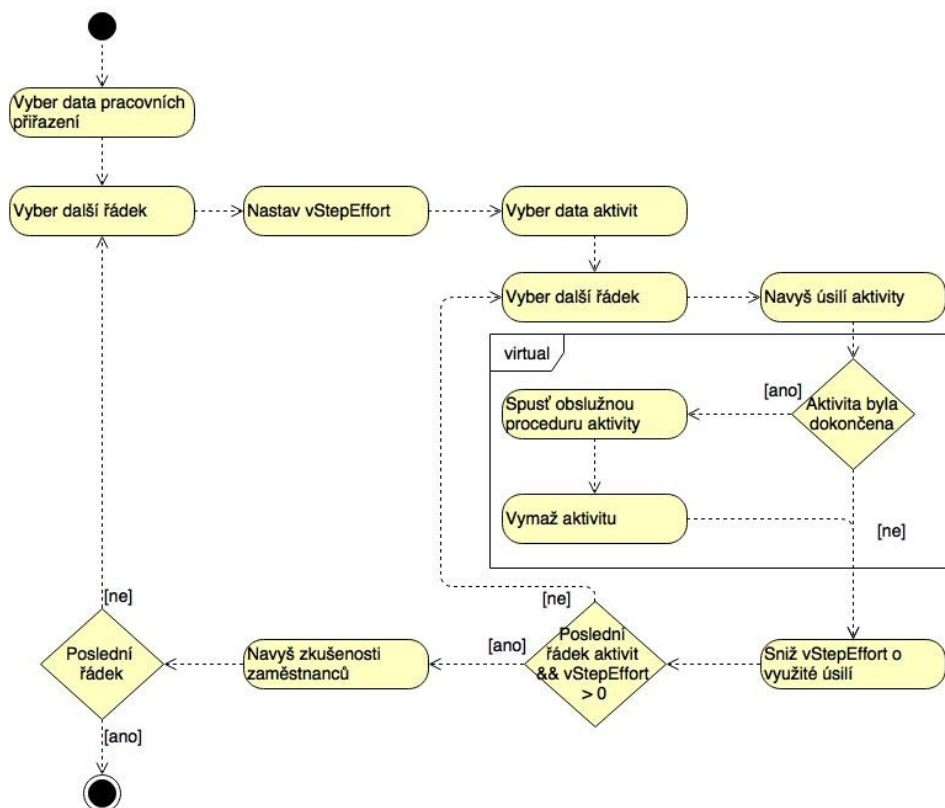
Proměnná `cQuantity` uchovává počet vystavených produktů v aktuálně procházeném prodejním regálu. Pokud je hodnota nulová, tak je provádění procedury pro aktuální regál přeskočeno.

6.2.2 Generování úsilí

Úsilí je generováno pro všechny pracovní přiřazení zaměstnanců k určitému vybavení. Po zjištění kolik úsilí tito zaměstnanci celkově vynaloží v jednom tahu, jsou postupně procházeny aktuálně rozpracované aktivity přidružené k tomuto vybavení a navyšován jejich stav k dokončení. Až dojde k vyčerpání hodnoty vynaloženého úsilí, je procházení aktivit přerušeno a dojde k navyšení zkušeností zaměstnanců.

Provádění procedury se liší s ohledem na typ aktivity. Chování většiny z nich může být zobecněno na spuštění příslušné obslužné procedury a následné vymazání aktivity, pokud byla tato

aktivita dokončena. Diagram aktivit níže (Obrázek 6.9) znázorňuje celou proceduru včetně zobecněného úseku označeného jako *virtual*.



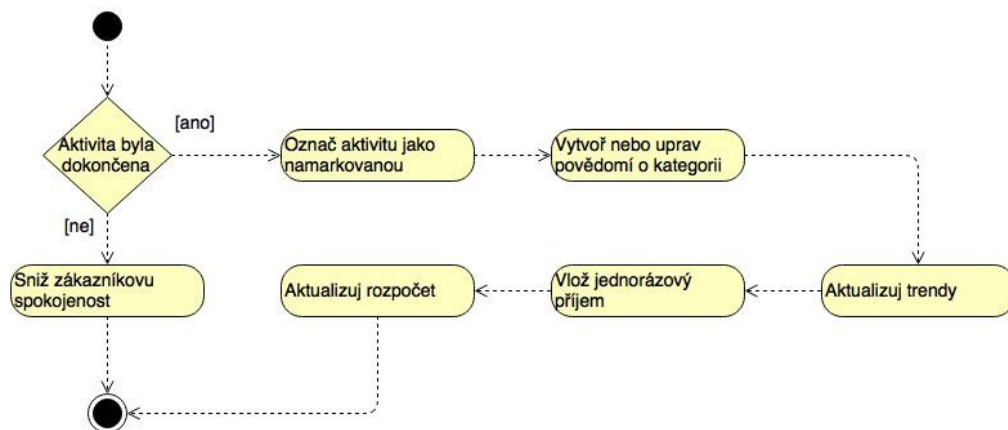
Obrázek 6.9 – Diagram aktivit pro generování úsilí

V diagramu je proměnná *vStepEffort* symbolem pro celkové množství vynaloženého úsilí všech zaměstnanců na určitém vybavení a pro určitý typ aktivity.

6.2.3 Generování úsilí na markování objednávek

Tato procedura konkretizuje proceduru pro generování úsilí popsanou výše, a to pouze v případě, že typ aktuálně procházené aktivity je markování objednávek. Pokud toto platí, je funkční blok, který byl výše označen jako *virtual*, nahrazen následující funkcionalitou.

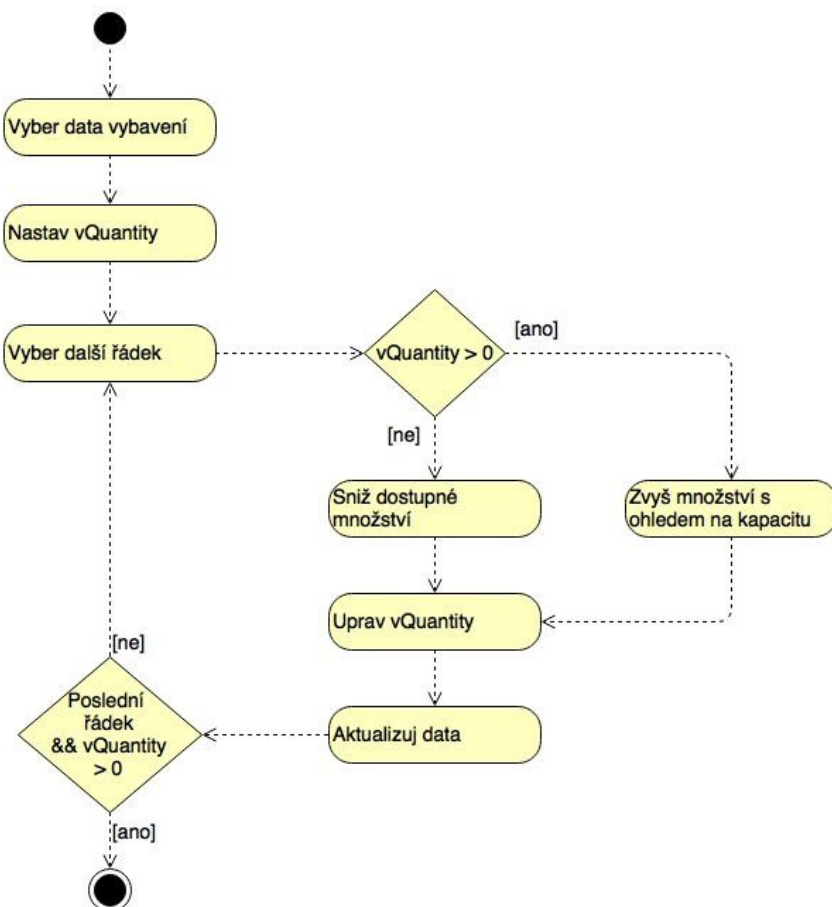
Pokud je aktivita markování objednávek dokončena, není ihned vymazána, nýbrž jen označena jako namarkovaná, tak aby uživatel mohl do vzniklých objednávek nahlížet. Následně dojde k aktualizaci obecného povědomí o kategorii příslušných produktů, trendů v regionu, ve kterém objednávka vznikla a rozpočtu společnosti, který bude navýšen o celkovou cenu zakoupených produktů. Pokud není markování dokončeno v jednom herním tahu, je snížena zákaznickova spokojenost dle nastavení v příslušném segmentu (viz Obrázek 6.10).



Obrázek 6.10 – Diagram aktivit pro generování úsilí na markování objednávek

6.2.4 Snižování a zvyšování množství zásob

Jelikož mohou být produkty naskladněny ve více skladovacích regálech s určitou kapacitou, je nutné pro změnu skladových zásob implementovat komplexnější funkcionalitu. Procedura prochází skladovací regály v nemovitosti sekvenčně a postupně mění stav zásob, dokud není přidáno nebo odebráno požadované množství nebo dokud neprojde všechny skladovací regály (viz Obrázek 6.11).

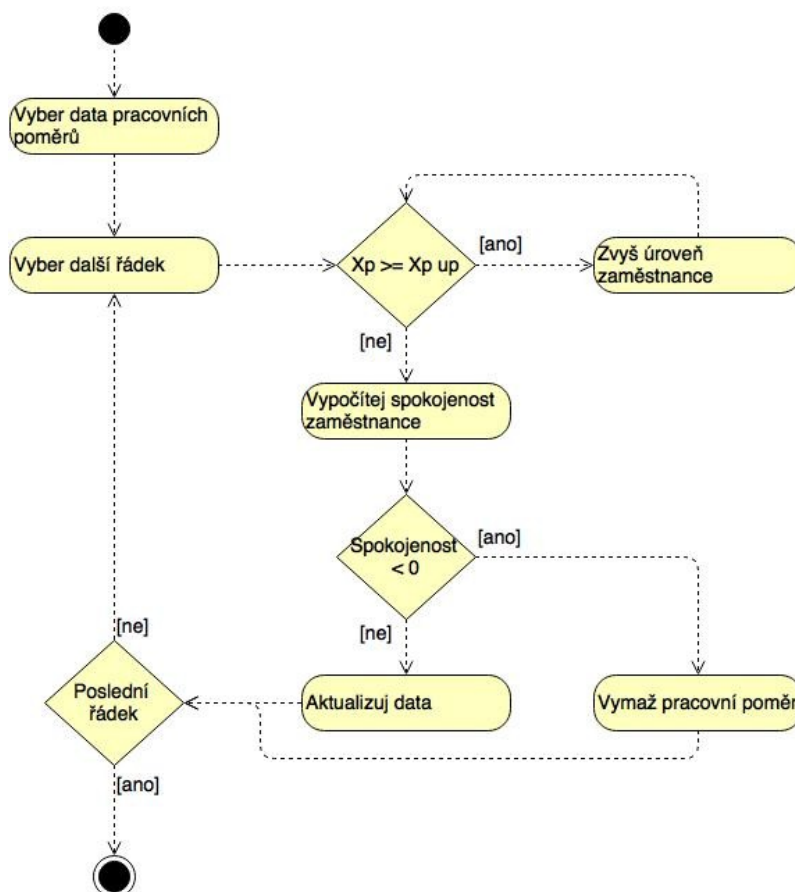


Obrázek 6.11 – Diagram aktivit pro snižování a zvyšování množství zásob

Toto množství je uchovááno v proměnné `vQuantity`. Pokud je hodnota větší než nula, je s každým průchodem množství ve skladovacím regálu navýšeno s ohledem na kapacitu tohoto regálu, a naopak pokud je menší než nula, je množství ve skladovacím regálu sníženo, je-li dostupné. Hodnota `vQuantity` je s každým průchodem aktualizována o množství, jež bylo opravdu ve skladovacím regálu změněno.

6.2.5 Aktualizace pracovních poměrů

Aktualizace pracovních poměrů spočívá v povyšování zaměstnance na vyšší úroveň a ve výpočtu jeho spokojenosti, na základě které může dojít i k podání výpovědi. Procedura prochází všechny pracovní poměry a zjišťuje, zda má náležitý zaměstnanec dostatek zkušeností k postupu na vyšší úroveň. Dále vypočítá a aktualizuje zaměstnancovu spokojenost na základě jeho úrovně, jeho pracovního vytížení a výši jeho platu. Pokud jeho spokojenost klesne do záporných hodnot, je pracovní poměr ukončen a vymazán z databáze. Obrázek 6.12 znázorňuje popisovanou funkcionalitu.



Obrázek 6.12 – Diagram aktivit pro aktualizaci pracovních poměrů

Proměnné `XP` a `XPUP` symbolizují množství zaměstnancových zkušeností a to jak jejich aktuální hodnotu, tak hodnotu potřebnou pro postup na vyšší úroveň.

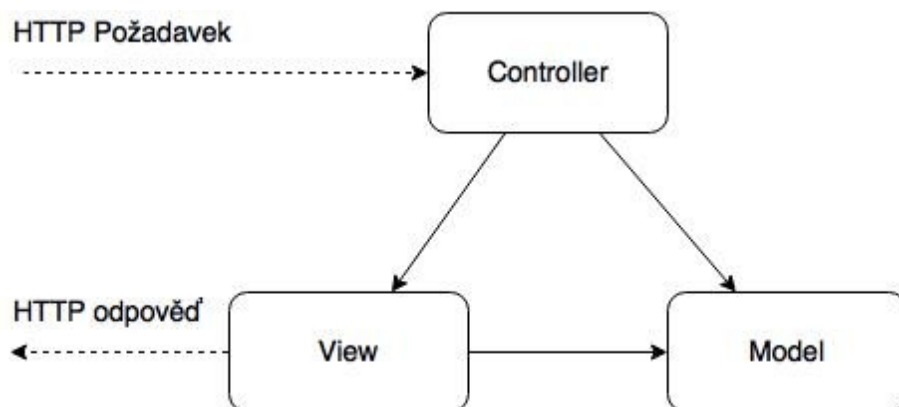
6.3 Serverová část aplikace

Serverová část aplikace je organizována a strukturována pomocí architektury MVC (z anglického model-view-controller). Termín byl poprvé použit v roce 1970 u projektu Smalltalk, který byl koncipován jako způsob rychlého vytváření uživatelského rozhraní aplikace. Některé detaily tehdy prezentované architektury MVC, byly uzpůsobeny jen pro projekt Smalltalk, nicméně základní myšlenka byla použitelná pro většinu vyvíjených aplikací, obzvláště těch webových [13].

Architektura odděluje doménovou logiku i doménový model od uživatelského grafického rozhraní aplikace, a utváří tak 3 samostatné části, jejichž názvy plynou přímo z pojmu MVC:

- **Model:** Reprezentuje nebo přímo uchovává samotná data. Pro model ovšem není striktně vyžadován určitý formát a prakticky je možné za model označit jakýkoliv objekt. V případě této aplikace se jako modely převážně používají objekty tříd popsaných v kapitole 6.1 případně pole nebo kolekce těchto objektů.
- **View:** Vykresluje daný model pomocí definované šablony a vytváří tak uživatelské rozhraní. U webových aplikací se šablona typicky vytváří pomocí HTML kódu.
- **Controller:** Směřuje uživatelské požadavky, zajišťuje požadovaná data a vybírá k nim náležící View pro jejich prezentaci.

Popsané části MVC architektury a jejich vzájemnou komunikaci znázorňuje Obrázek 6.13.



Obrázek 6.13 – Architektura MVC

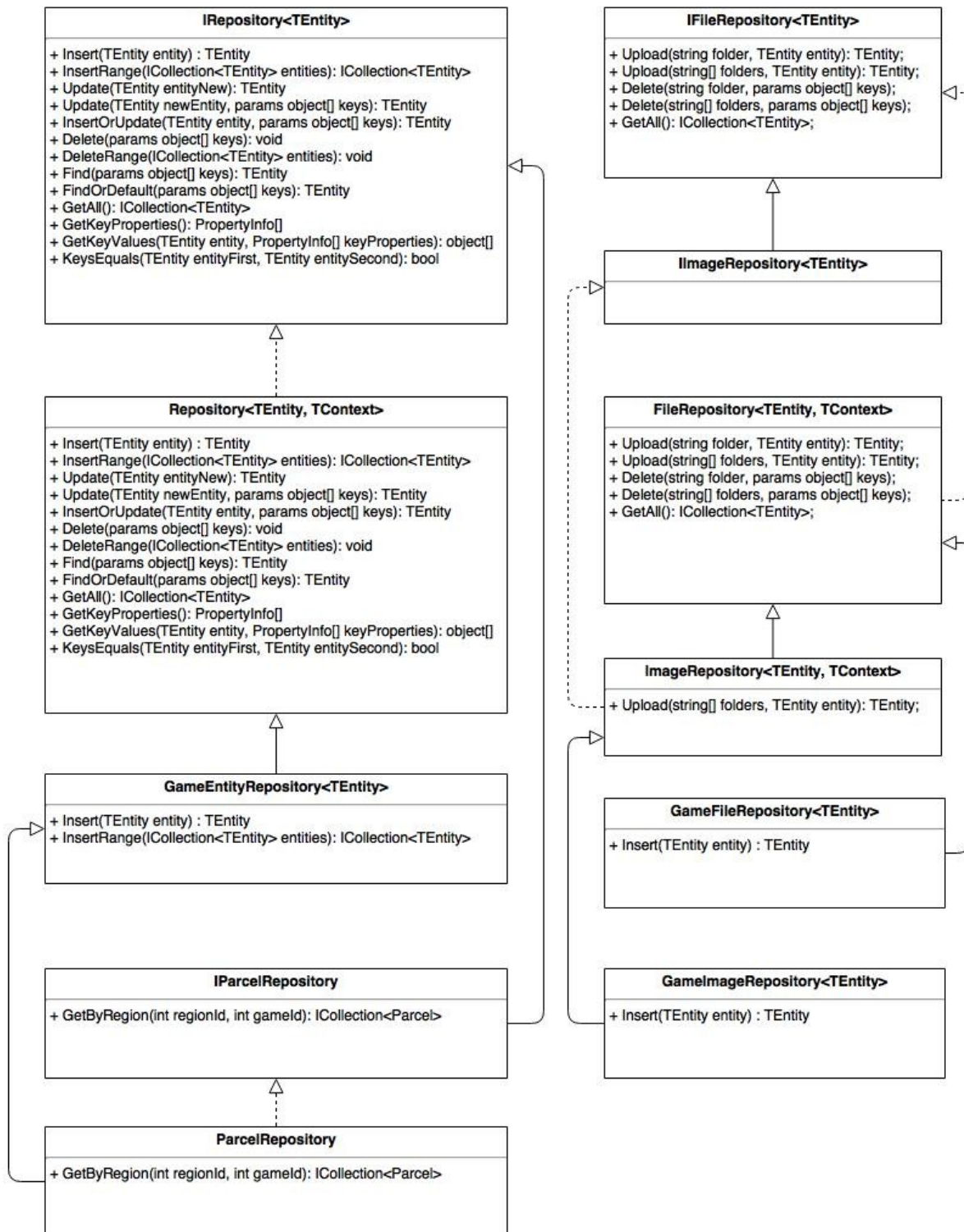
6.3.1 Přístup k datům

U rozsáhlejších aplikací je výhodné jednak rozlišovat logiku a prezentační vrstvu a jednak vrstvu přístupu k datům. Ta tvoří prostředníka mezi logikou aplikace a datovým zdrojem. Poskytuje tedy rozhraní s metodami zajišťující jak standardní CRUD operace, tak například třídění nebo složitější selekci dat. Pokud je datovým zdrojem relační databáze, tak musí zajišťovat i objektově – relační mapování.

Důležitým návrhovým vzorem řešícím tuto problematiku je `Repository`. Ten pro určitou třídu doménového modelu používá mapující objekt, který právě nad touto třídou, zajišťuje popisovanou funkcionalitu. Pro standardní metody je možné vytvořit generické rozhraní. V této

aplikaci je to rozhraní `IRepository`, které je použito pro většinu tříd doménového modelu. Výjimkou jsou třídy odvozené od `File`, se kterými pracuje rozhraní `IFileRepository` respektive `IImageRepository`. Tyto třídy jsou totiž mapovány na datový zdroj, ale i na souborový systém, kde mají přidružen určitý soubor, respektive obrázek. Konkrétní implementace `IFileRepository` se tedy musí starat i o nahrání a smazání souboru ze souborového systému. Implementace `IImageRepository` pak předpokládá, že přidružený soubor je obrázek, a doplňuje funkcionalitu `IFileRepository` o možnost zmenšování obrázku a vytváření miniatur.

Obrázek 6.14 znázorňuje strukturu tříd `Repository` s ukázkou implementace koncového mapujícího objektu pro třídu `Parcel`, tedy `ParcelRepository`.



Obrázek 6.14 – Třídní diagram struktury repositářů

Generické mapující třídy `GameEntityRepository`, `GameFileRepository` a `GameImageRepository` pracují s třídami `GameEntity`, `GameFile` a `GameImage` a starají se o korektní inkrementaci `Id` a o aktualizaci času poslední úpravy.

6.3.2 Služby

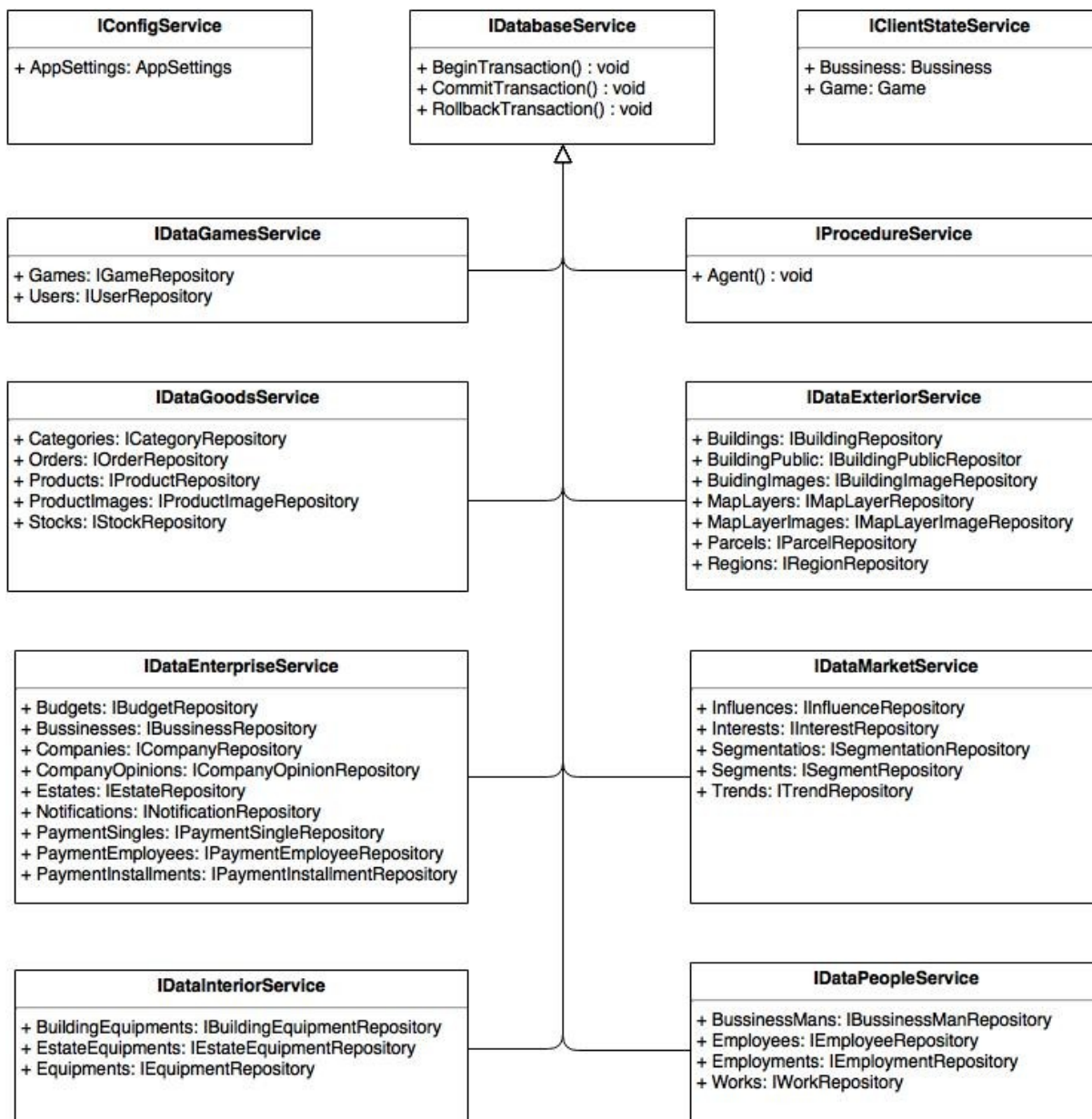
Každá aplikace, aby mohla být snadno rozšířitelná a udržitelná, musí být co nejméně závislá na konkrétních implementacích. Již v roce 1995 při vzniku návrhových vzorů GOF byl doporučován princip vytváření závislosti proti rozhraní, nikoliv proti implementaci.

Jedním ze softwarových řešení jak tohoto dosáhnout je vkládání respektive vstřikování požadovaných závislostí (z anglického dependency injection). Použitím DI, vzniká možnost, pro určitou funkcionalitu zde označovanou jako služba, jednoduše zaměnit konkrétní implementace. To je výhodné například pro testování, nebo pro přístup k různým datovým zdrojům a podobně. Ovšem pravá síla DI, se projeví až při vývoji rozsáhlejších aplikací.

Pro správu služeb a řízení jejich životního cyklu může sloužit DI kontejner. Ten je nutné před použitím v aplikaci nakonfigurovat a přiřadit jednotlivým službám požadované implementace. Pokud si poté aplikace vyžádá instanci některé služby, DI kontejner se postará o její zprostředkování. DI kontejner není nezbytný pro fungování samotného DI, ovšem bez jeho použití, je nutné zajistit vytváření instancí manuálně.

Je více možností jak s požadovanou službou pracovat. Standardně by se služba měla předávat přímo pomocí konstruktoru. Tímto způsobem je zajištěna existence závislosti po celou dobu existence třídy, která s ní pracuje. Další možností je dodat instanci služby až do vlastnosti dané třídy. Je zřejmé, že tato vlastnost musí být veřejně přístupná jak pro čtení, tak pro zápis. Proto by se měl tento přístup používat pouze v případě, kdy je existence služby nepovinná. Třetím a posledním způsobem je dodání závislé služby jako parametru určité metody. To umožňuje pro každé volání metody zajistit různé instance služeb [11].

Služby dostupné v této aplikaci zobrazuje třídní diagram níže.



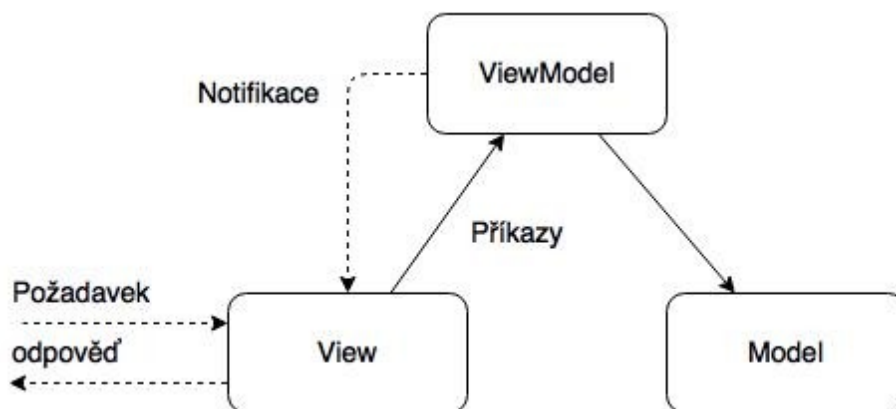
Obrázek 6.15 – Třídni diagram služeb

Služba **IConfigService** slouží k uchování nastavení aplikace, zatímco služba **IClientStateService** slouží k uchování klientských informací, což zahrnuje uložení aktuálně vybrané hry a aktuálně vybrané společnosti respektive fyzické osoby v rámci této hry. Služba **IDatabaseService** působí jako předek všech služeb pracujících s databází a poskytuje metody pro práci s transakcemi. Rozšiřující služby s prefixem **IData** zpřístupňují jednotlivé repozitáře uskupené dle odpovídajících částí doménového modelu. Poslední službou je **IProcedureService**, jejichž metody jsou mapovány na databázové procedury, přičemž služba zajišťuje jejich korektní volání.

6.4 Klientská část aplikace

Značná funkcionality aplikace je předpokládána i na klientské straně. Proto se i zde budou skripty organizovat do určité architektury. Tou architekturou je MVVM (z anglického model-view-viewmodel).

MVVM z velké části vychází z architektury MVC. Napovídá tomu i fakt, že písmena MV mají obě architektury společné a se stejným významem, tedy model a view. VM znamená viewmodel, což je v kontextu MVVM objekt, jež uchovává stav aplikace, zapouzdřuje její funkce a podobně jako MVC controller, spojuje model a view (viz Obrázek 6.16).

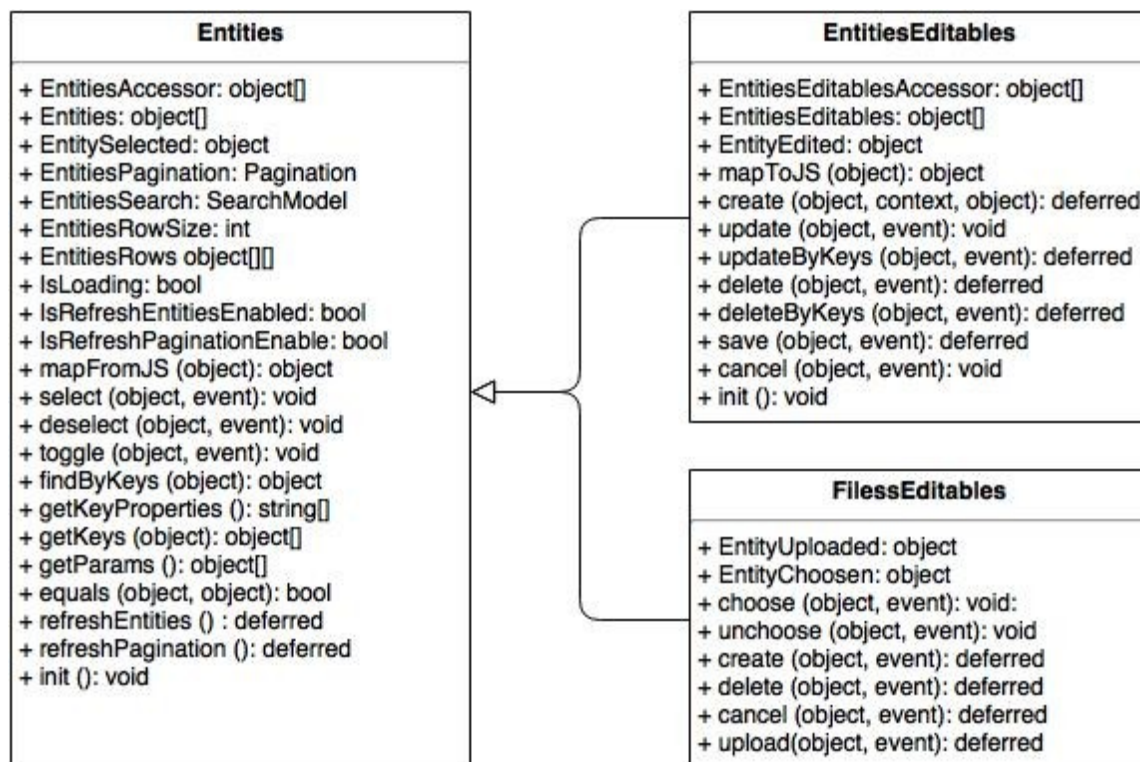


Obrázek 6.16 – Architektura MVVM

Principem vývoje aplikace v architektuře MVVM je deklarativní definice oboustranného svázání vlastností a metod viewmodelu s určitým view. Pokud je tedy programově změněn stav viewmodelu, změny se automaticky projeví i v grafickém rozhraní a naopak, pokud uživatel provede nějakou interakci, tak je aktualizován stav viewmodelu. Opouští se tedy od událostmi řízeného programování. Obsluhu událostí řeší MVVM framework interně.

Jednou z výhod architektury je možnost strukturalizace viewmodelů. Pro klientskou část této aplikace bude použit podobný přístup jako při tvorbě MVC controllerů, které jsou vytvářeny jednotlivě pro třídy doménového modelu. Tady bude možno mít více viewmodelů operujících s konkrétní doménovou třídou, a to proto, aby bylo možné pracovat s různými selekcemi dat. Je zřejmé, že tato struktura obsahuje hodně společné funkcionality, nicméně je možné použít generalizaci a implementovat společné části v nadřazených třídách¹ (viz Obrázek 6.17).

¹ Třídy jako takové se u klientských jazyků nevyskytují. Termín je použit jen v rámci návrhu.



Obrázek 6.17 – Třídní diagram struktury viewmodelů

Třída `Entities` poskytuje způsob jak načíst data, převést data na objekty a ty potom zobrazit v grafickém rozhraní s možností interaktivního stránkování, vyhledávání a výběru. Způsob mapování vstupních dat na objekty je dán metodou `mapFromJS`. Objekty jsou uchovány ve vlastnosti `Entities`, přičemž o jejich aktualizaci se v případě potřeby stará metoda `refreshEntities`. Vlastnost `EntitiesPagination` uchovává model potřebný pro zobrazení ovládacích prvků stránkování, přičemž je k ní rovněž přidružena aktualizací metoda `refreshPagination`. Obě aktualizací metody lze povolit nebo zakázat pomocí nastavení binárních vlastností `IsRefreshEntitiesEnabled` a `IsRefreshPaginationEnabled`. Pro vyhledávání je potřebná vlastnost `EntitiesSearch` uchovávající náležitý model s potřebnými informacemi, zatímco pro výběr je důležitá vlastnost `EntitySelected`, která uchovává aktuálně vybraný objekt a ten lze pomocí metod `select`, `deselect` a `toggle` nastavovat.

Pokud je třeba objekty zobrazit v tabulce, je možné využít vlastnosti `EntitiesRows`, která mapuje objekty z vlastnosti `Entities` na pole polí těchto objektů. Velikost vnějších polí, je dána vlastností `EntitiesRowSize`, která tedy reprezentuje počet objektů v jednom řádku tabulky.

Třída `EntitiesEditables` přidává ke standardním zobrazovacím operacím ještě možnost interaktivní editace objektů, respektive přidání objektu, úpravu zvoleného objektu, uložení provedených úprav a smazání zvoleného objektu. Vytvoření nového objektu je zajištěno metodou `create`, která si u serveru vyžádá data popisující jeho strukturu a ty převede na objekt. Ten následně označí jako upravovaný, když ho uloží do vlastnosti `EntityEdited`. Metody `update` a `updateByKey` taktéž nastavují vlastnost `EntityEdited`, ovšem použijí k tomu existující

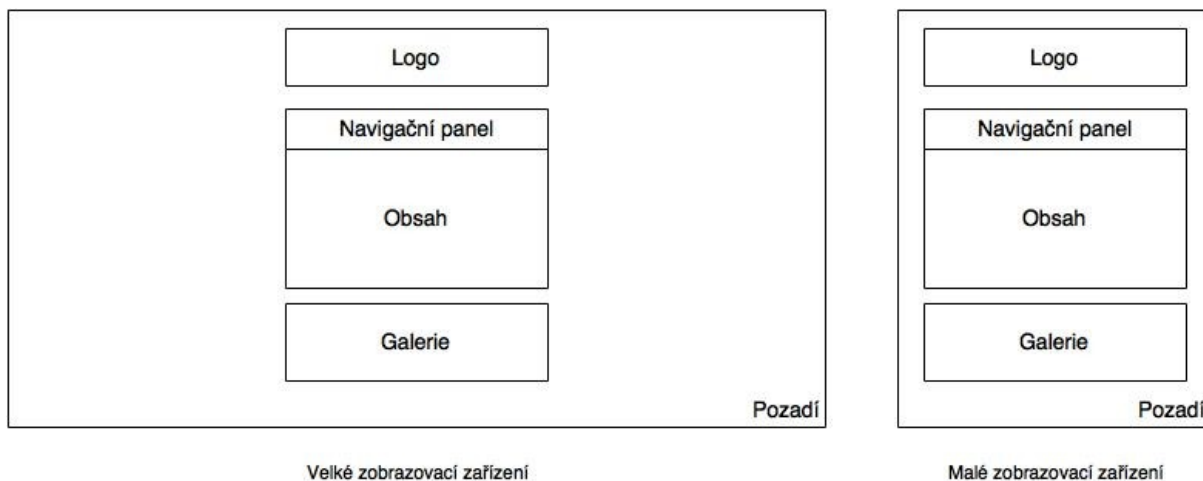
objekty v prvním případě předané parametrem a ve druhém případě načtené ze serveru. Podobně je tomu i u metod `delete` a `deleteByKey`. Ty ovšem slouží ke smazání existujících objektů. Uložení upravovaného objektu je v režii metody `save`. Ta ho pomocí metody `mapToJS` převede zpět na data a pošle požadavek na server.

Poslední třídou v diagramu je třída `FilesEditables`, která je primárně použita pro svázání s modálním oknem pro zvolení souboru eventuálně jeho nahrání z lokálního počítače. K těmto účelům jsou přidány vlastnosti `EntityChosen` nesoucí zvolený soubor a `EntityUploaded` reprezentující právě nahrávaný soubor. Metoda `create` v tomto případě nastavuje vlastnost `EntityUploaded`, a o samotné nahrání souboru se stará metoda `upload`. Zatímco o práci s `EntityChosen` se starají metody `choose` a `unchoose`.

6.5 Grafické rozhraní

Při návrhu rozvržení grafických prvků je nutné brát v úvahu požadavek na responzivní design i na možnost interakce jak pomocí klávesnice a myši tak i pomocí dotyku. Většina vstupních formulářů, ale i výstupních informačních bloků, bude řešena pomocí moderních modálních oken. Rozvržení základních stránek bude v rámci určitých části aplikace téměř ekvivalentní.

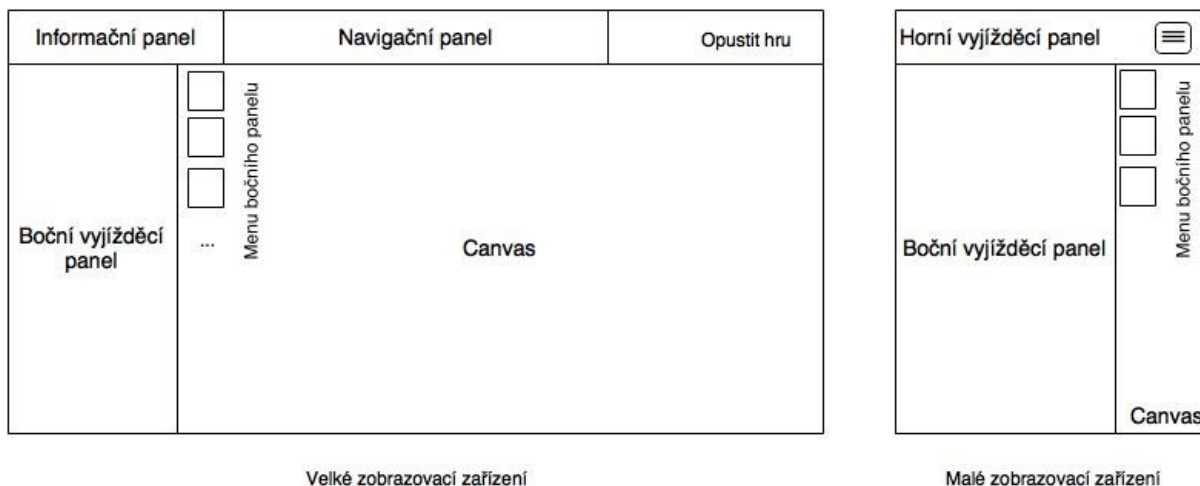
Výchozí zobrazení bude uživateli prezentovat pouze nezbytné informace o aplikaci a poskytovat interakční možnosti pro provádění některých případů užití výchozí části aplikace (viz 5.3.1). Grafické prvky budou soustředěny na střed obrazovky, přičemž budou obklopeny obrázkovým pozadím. Obsah stránek bude vykreslován pod logem webu a nad obrázkovou galerií, poskytující rychlý náhled na aplikaci. Obsah bude měněn pomocí výše umístěného navigačního panelu. Obrázek 6.18 ukazuje rozvržení grafických prvků na velkém i malém zobrazovacím zařízení.



Obrázek 6.18 – Grafické rozhraní – výchozí část

Samotné herní prostředí bude vykreslováno v HTML elementu `Canvas`, který bude vyplňovat téměř celou oblast obrazovky. Pro rychlý informační přehled a navigaci v aplikaci je k dispozici horní panel, který bude na malém zobrazovacím zařízení vyjíždějící. Vyjíždějící je i boční panel a to

u zařízení jakékoliv velikosti. Obsah bočního panelu je možné měnit pomocí ikon v menu bočního panelu (viz Obrázek 6.19).

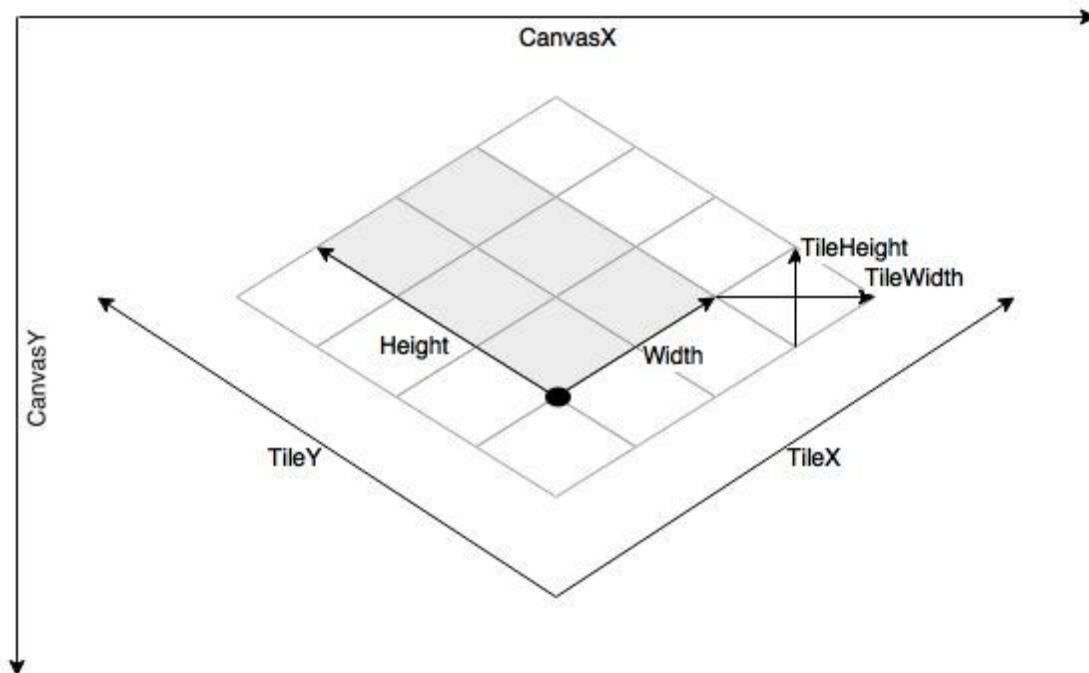


Obrázek 6.19 – Grafické rozhraní – herní uživatelská část

V popisovaném Canvasu je vykreslována jednak mapa prostředí a jednak i izometrické mapy regionu a nemovitosti.

6.5.1 Izometrická mapa

Mapa je tvořena kosočtvercovým uskupením izometrických objektů, jež jsou vykresleny pomocí náležitých obrázků. Objekty nemusí být stejně rozměrné jako jedno pole mapy, nýbrž mohou těchto polí pokrýt více. Obrázek 6.20 ukazuje příklad mapy, kde je šedou barvou vyznačen izometrický objekt.



Obrázek 6.20 – Izometrická mapa

Každý objekt má definovanou svou šířku (*Width*) na ose *x* a výšku (*Height*) na ose *y* a také své souřadnice (*TileX*, *TileY*). Všechny tyto údaje jsou udány v izometrickém souřadnicovém systému. Aby mohla být mapa korektně vykreslena, musí dojít k převodu souřadnic na souřadnice používané v elementu Canvas. K tomu je nutné znát rozměry jednoho pole mapy (*TileWidth*, *TileHeight*) [12]. Rovnice 6.1 ukazuje způsob realizace tohoto převodu.

$$CanvasX = TileX * \frac{TileWidth}{2} - TileY * \frac{TileWidth}{2}$$

$$CanvasY = TileX * \frac{TileHeight}{2} + TileY * \frac{TileHeight}{2} * -1$$

TileX, *TileY* – souřadnice objektu

TileWidth, *TileHeight* – rozměry jednoho pole mapy

Rovnice 6.1 – Převod izometrických souřadnic

V aplikaci se mapa zobrazuje v závislosti na bodu představující střed mapy. Změnou polohy tohoto středu, dojde k vykreslení jiné části mapy. Tímto způsobem je implementována funkce interaktivního posunu.

6.5.2 Logo

Aplikace byla pojmenována smyšleným anglickým názvem Trade Match. V českém jazyce toto slovní spojení znamená „obchodní zápas“. Při návrhu loga byla snaha o vystižení ekonomické tematiky. Výsledné logo je k vidění na obrázku dole.



Obrázek 6.21 – Logo

Tvar v levé části je kombinací symbolu amerického dolaru a symetrických šipek, jež v tomto kontextu symbolizují obchod.

7 Implementace

Kapitola je věnována samotné implementaci navržené aplikace. Jsou zde stručně charakterizovány rysy použitých technologií a popsána organizace zdrojového kódu v rámci aplikace.

7.1 Použité technologie

Aplikace byla na serverové straně implementována v jazyce C# s využitím .NET Frameworku 4.5. Pro prezentační vrstvu byl použit značkovací jazyk HTML 5 v kombinaci s kaskádovými styly CSS 3. Logika klientské části byla vytvořena pomocí javascriptu s využitím některých dnes již standardně používaných knihoven (například jQuery).

Jako vývojové prostředí bylo zvoleno Visual Studio 2013 Community, které je k dispozici zdarma i pro komerční použití.

7.1.1 ASP.NET MVC 5

Platforma ASP.NET byla původně určena pro použití s technologií Web Forms. Podpora pro MVC byla přidána později. Mimo výhody, jež plynou z použité architektury MVC (viz kapitola 6.3) je třeba zmínit i možnost úplné kontroly vykresleného HTML kódu, snadné obsluhy asynchronních požadavků a jednoduché integrace javascriptových knihoven a frameworků [13].

Právě kvůli rozsáhlé klientské logice a nutnosti implementace vlastního API, obsluhujícího asynchronní operace, byla pro serverovou část aplikace zvolena technologie ASP.NET MVC 5.

7.1.2 Microsoft SQL Server

SQL Server je nástroj navržený pro řízení, zabezpečení a poskytování přístupu k datům, uložených jako relační struktury v databázích. Se SQL Serverem může vývojář a administrátor komunikovat pomocí jazyka SQL, který jednak slouží pro definici dat (DDL), a jednak pro manipulaci s daty (DML). Přímým nástupcem jazyka SQL je jazyk T-SQL, který přináší možnost procedurálního programování [14].

Díky T-SQL mohla být systémová část této aplikace implementována na databázovém serveru což je přínos hlavně po výkonové stránce.

7.1.3 Entity framework

Entity framework je jedním z nástrojů řešících objektově relační mapování, tedy proces přeměny relačních databázových struktur na objektový model, jež je použitelný v objektově orientovaném programovacím jazyce. Entity framework poskytuje dva základní přístupy k mapování. Jednak lze existující databázi mapovat na automaticky vytvářený objektový model (database-first) nebo naopak manuálně vytvořený objektový model mapovat na automaticky synchronizovanou databázi (code-first) [14]. Právě druhý přístup je použitý v této aplikaci.

7.1.4 KnockoutJS

Javascriptová knihovna KnockoutJS je určená pro tvorbu dynamických uživatelských rozhraní webových aplikací. KnockoutJS je postaven na architektuře MVVM (viz 6.4) a slouží výhradně ke svázání uživatelského rozhraní s modelem. Nesnaží se tedy být komplexním

frameworkem poskytující všechnu potřebnou funkcionalitu. Knihovna je k dispozici zdarma a je jednou z nejmenších z dostupných knihoven a frameworků, které tuto problematiku řeší [16].

7.1.5 EaselJS

Knihovna EaselJS poskytuje objektové rozhraní pro práci s vykreslovanou 2D grafikou v HTML5 elementu Canvas. Umožňuje s vykreslenými obrázky a geometrickými tvary pracovat jako s objekty, jež vytvářejí hierarchickou strukturu a jsou schopné reagovat na uživatelskou interakci. EaselJS je samostatně použitelná součást komplexního řešení CreateJS [17].

Aplikace používá EaselJS pro vykreslování mapy prostředí, mapy regionu i mapy nemovitosti.

7.1.6 Bootstrap

Ve firmě Twitter byl požadavek na standardizaci nástrojů pro tvorbu grafického rozhraní webové aplikace. V srpnu roku 2011 byl pro veřejnost vypuštěn framework Bootstrap a ihned si získal svou popularitu. Bootstrap prošel vývojem od čistě CSS projektu až ke komplexnímu řešení, které v sobě nese i skripty, ikony a podobně. Pomocí Bootstrapu může vývojář lehce vytvořit responzivní webový design obsahující různé uživatelsky přívětivé prvky jako jsou modální okna a rozbalovací nabídky [18].

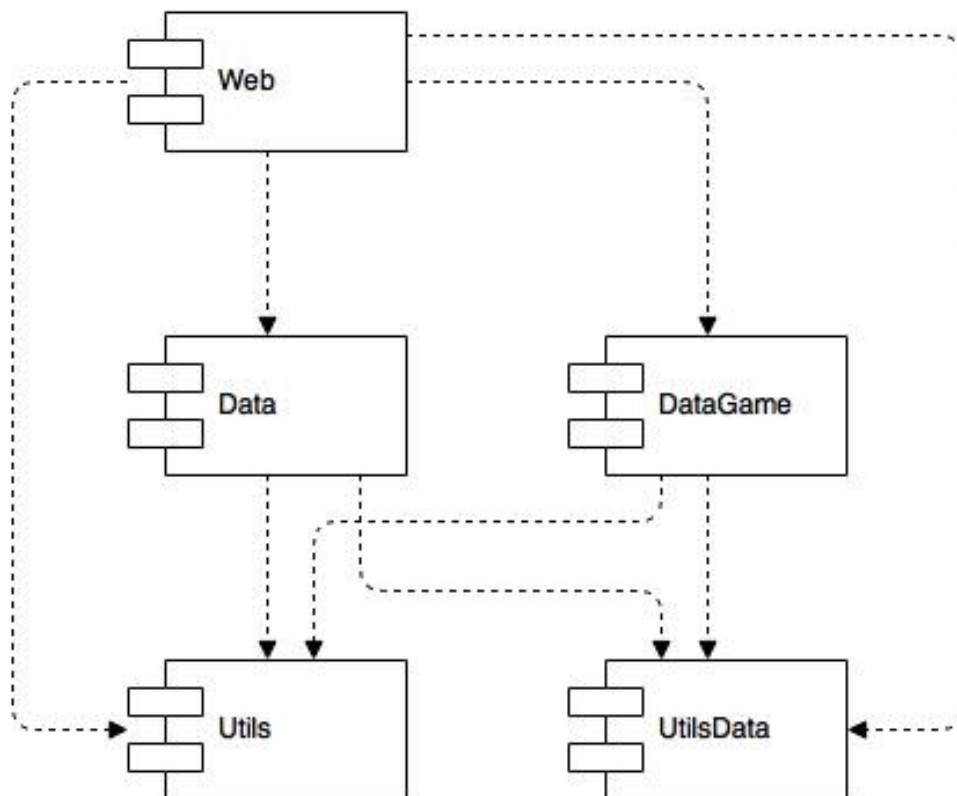
7.1.7 Blender

Blender je zdarma dostupný 3D modelovací nástroj. Byl použit pro modelování herních objektů, což umožnilo jejich vykreslení z libovolného úhlu a v libovolné velikosti. Zdrojové soubory ve formátu blend jsou součástí výsledného řešení.

7.2 Komponenty

Aplikace je rozdělena do několika komponent, kdy každá slouží jinému účelu. Jejich vzájemné závislosti ukazuje Obrázek 7.1.

- Web: Komponenta představuje výchozí bod celé aplikace. Je zodpovědná za obsluhu synchronních i asynchronních HTTP požadavků a obsahuje základní serverovou logiku a prezentační vrstvu.
- Data: Řeší přístup k datům, které nejsou závislé na žádné instanci hry.
- DataGame: Řeší přístup k datům, které se vztahují k nějaké konkrétní instanci hry.
- Utils: Jedná se o generickou znovupoužitelnou komponentu obsahující pomocnou logiku.
- UtilsData: Obdobně jako komponenta Utils je i komponenta UtilsData generická a znovupoužitelná, ovšem její pomocná logika se soustřeďuje na datovou vrstvu.



Obrázek 7.1 – Diagram komponent

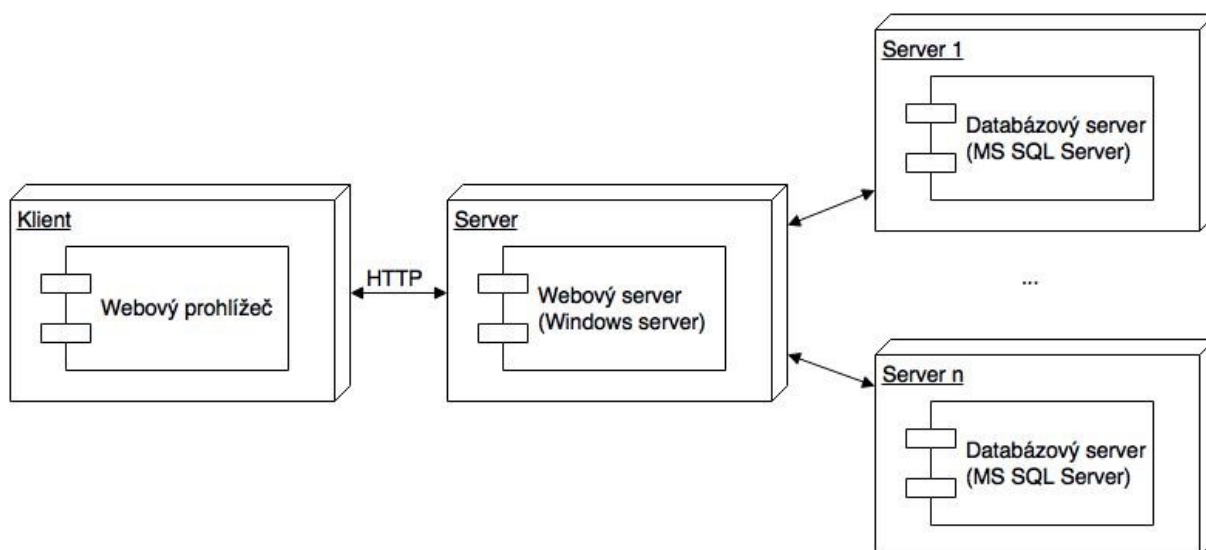
7.3 Struktura webové aplikace

MVC webová aplikace v komponentě Web je strukturována modulárně. Každý modul pracuje se svými modely, controllery a view a je zasazen do samostatného adresáře. Výchozí modul je umístěn v kořenovém adresáři aplikace a je zodpovědný za vykonávání případů užití popsaných v kapitole 5.3.1. Ostatní dílčí moduly jsou popsány níže:

- Play: Modul poskytuje přístup k herní instanci z pohledu uživatele - hráče. Vykonává případy užití herní uživatelské části (viz 5.3.2).
- Admin: Stejně jako předchozí modul poskytuje přístup k herní instanci, ovšem z pohledu administrátora. Stará se tedy o vykonání případu užití administrační části (viz 5.3.4).
- Api: Controllery tohoto modulu se nestarají o vykreslování stránek a neobsahují žádnou klientskou logiku, pouze zajišťují přístup k datům a obrázkům. Data jsou vráceny ve formátu JSON.

7.4 Diagram nasazení

Použité hardwarové prostředky, jež byly použity pro nasazení, ukazuje Obrázek 7.2. Klient pomocí webového prohlížeče přistupuje k webovému serveru a ten potom k jednomu nebo více databázovým serverům. Aplikace byla navržena tak, aby každá herní instance mohla pracovat s jiným databázovým serverem a tedy i s jinou databází, aby bylo docíleno lepšího výkonu.



Obrázek 7.2 – Diagram nasazení

8 Výsledky a testování

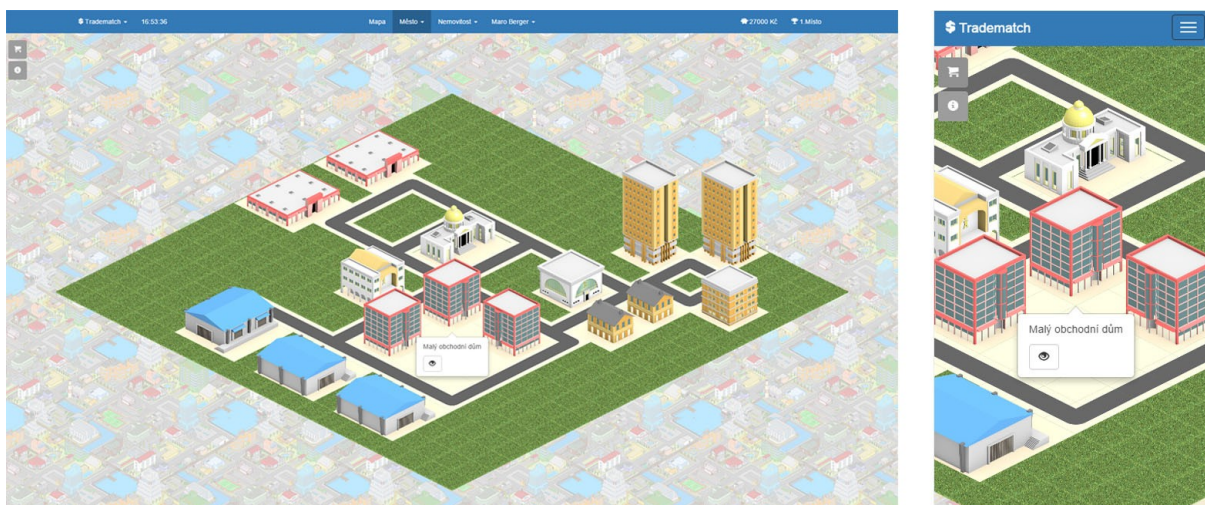
Poslední kapitola této diplomové práce poskytuje náhled vyhotovené aplikace, zmiňuje průběh testování a popisuje možnosti budoucího rozšíření.

Obrázek 8.1 zobrazuje výchozí bod aplikace po autorizaci a autentizaci, v podobě rozhraní pro výběr požadované herní instance. Na pravé straně je zde, i na následujících obrázcích této kapitoly, uvedeno zobrazení na zařízeních s malým displejem.



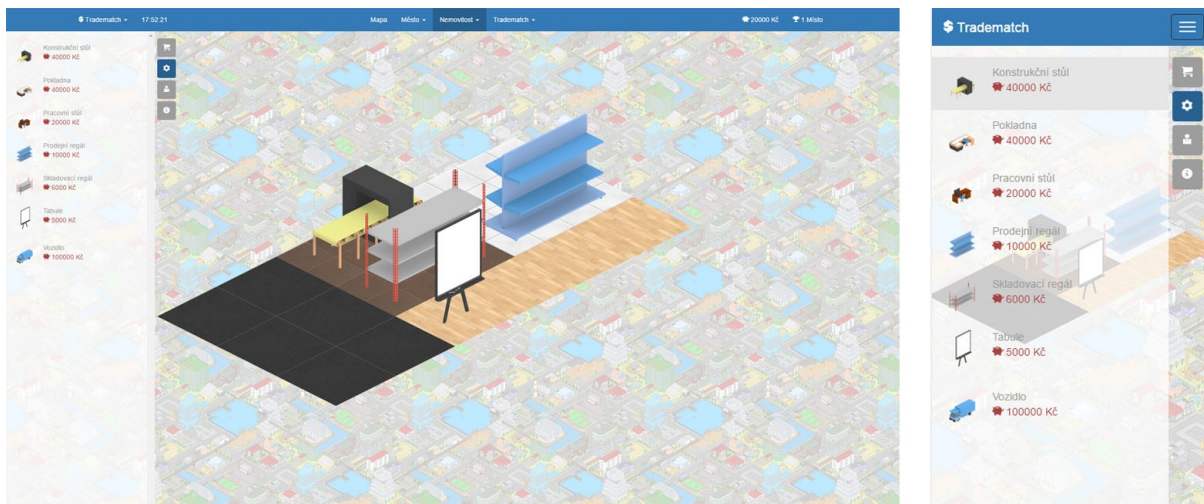
Obrázek 8.1 – Výběr hry

Po volbě hry je uživatel přesměrován na mapu prostředí s možností volby regionu, kdy je po jejím provedení zobrazena příslušná mapa (Obrázek 8.2). Pohyb po mapě je možný jak myší tak i dotykem prstu.



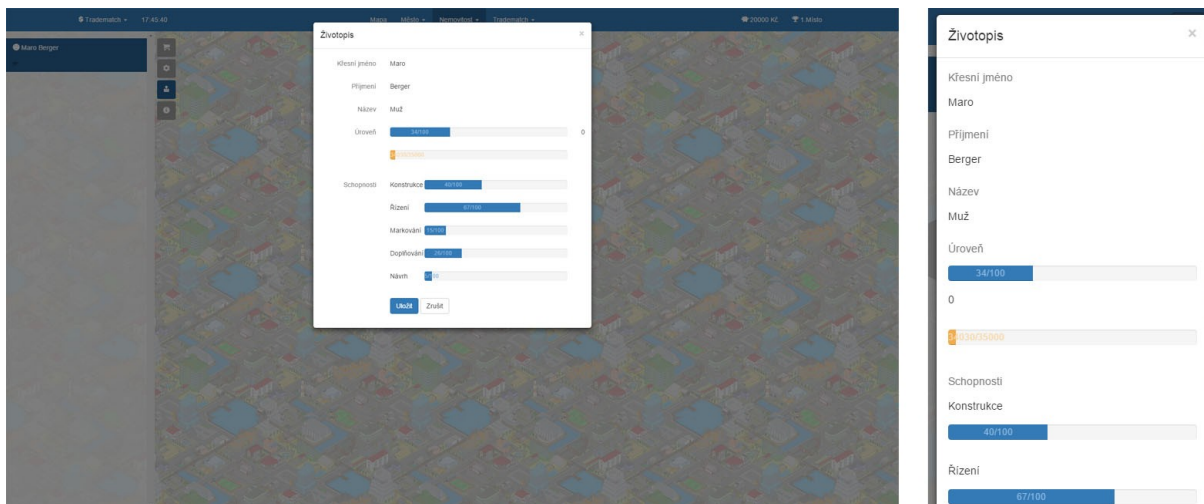
Obrázek 8.2 – Mapa regionu

Obdobným způsobem je vykreslována i mapa nemovitosti (Obrázek 8.3). Úprava určitého vybavení je možná po jeho selekci, kdy se pod ním zobrazí informační okno s příslušnými editačními možnostmi. Nové vybavení je možné vybírat z levého rozbalovacího panelu.



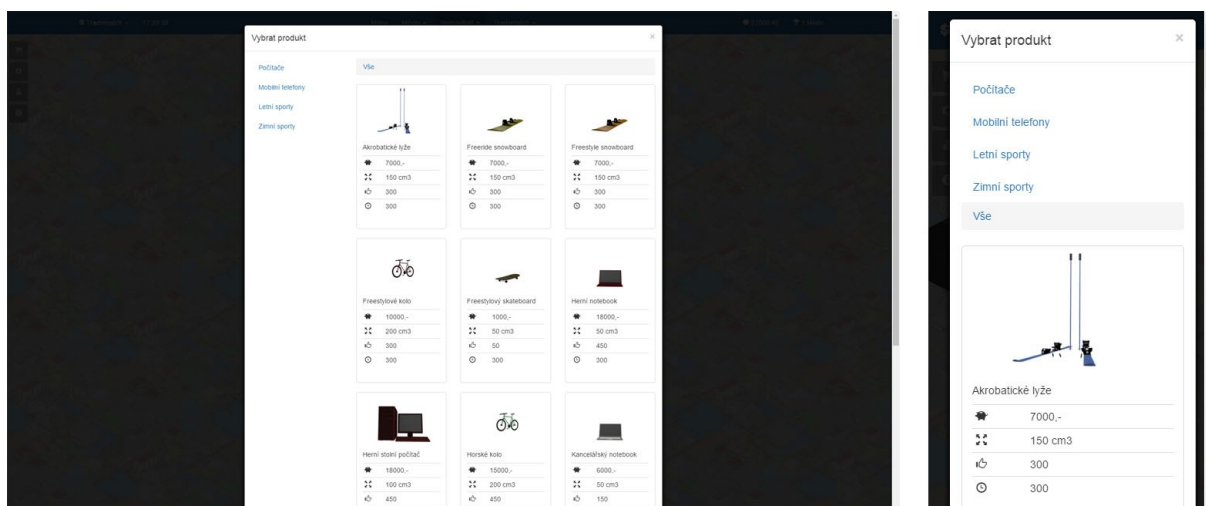
Obrázek 8.3 – Mapa nemovitosti

Kromě vybavení zobrazuje levý panel i dostupné zaměstnance společnosti. Jejich životopisy, včetně možností vylepšování schopností, jsou poskytovány pomocí modálního okna, které uvádí Obrázek 8.4.



Obrázek 8.4 – Okno zaměstnance

Obdobná modální okna jsou v aplikaci použita u většiny interaktivních funkčních bloků. Velká část z nich vyžaduje po uživateli, aby vybral produkt. Samotný výběr je řešen dalším modálním oknem, jež poskytuje detailní informace o každém z nich (Obrázek 8.5).



Obrázek 8.5 – Okno výběru produktu

8.1 Možnosti budoucího rozšíření

V následujícím textu jsou popsány potencionální možnosti pro rozšíření implementované aplikace.

8.1.1 Rozšíření sortimentu

Aplikace umožňuje uživateli do svého sortimentu zahrnovat pouze předdefinované produkty. Rozšiřováním této předdefinované nabídky by mohlo dojít k zaujmutí dalších uživatelských segmentů.

Většinu reálných produktů lze jistým způsobem rozdělit na množinu dílčích komponent, ať už jde o produkty z kategorie výpočetní techniky nebo sportu. Například počítač se skládá z procesoru, paměti, pevného disku a tak dále, zatímco například in-line brusle se skládají z boty, rámu a koleček. Pokud by byla aplikace doplněna i o komponenty, byla by do jisté míry vytvořena iluze svobodného návrhu vlastního produktu, kde by si uživatel mohl vybrat, z čeho se produkt bude skládat a s každým výběrem ovlivňovat výsledný vzhled i charakter produktu. Výsledek by pak mohl uvést na trh pod vlastní značkou.

Aplikace by kromě produktového sortimentu mohla umožňovat i nabízení služeb. Uživatel by tak mohl mít možnost vytvořit například fitcentrum, kam by zákazníci chodili posilovat.

8.1.2 Alternativní motivy

Aplikace je striktně zasazena do městského prostředí současné doby. Doménový model je ovšem připraven na vytváření vlastního motivu a schází pouze dopracovat administraci budov a vybavení. Díky této funkcionalitě by pak prostředí hry mohlo být zasazeno do jakéhokoliv prostředí, ať už by šlo například o středověk nebo o western a podobně. Alternativnímu motivu samozřejmě může být uzpůsoben i produktový sortiment.

8.1.3 Mobilní aplikace

Díky implementaci mobilní aplikace pro operační systémy Android, Windows Phone a iOS, jež by byla s webovou aplikací propojena, by bylo docíleno přívětivějšího rozhraní, než jaké je nyní vytvářeno responzivním designem ve webovém prohlížeči. Umístěním mobilní aplikace na náležitá úložiště jednotlivých platforem by mohly být osloveny další segmenty uživatelů.

8.1.4 Placené služby

Podobně jako ve hře Travian (viz 3.4) a jí podobných, se i zde nabízí možnost poskytovat uživatelům zpoplatněné zvýhodňující služby. Díky nim, může uživatel získat konkurenční výhodu oproti ostatním, a to například navýšením rozpočtu nebo dočasným urychlením činnosti zaměstnanců a podobně.

8.2 Testování

Pro účely testování i budoucího nasazení aplikace byla zakoupena doména www.tradematch.cz společně se sdíleným webovým virtuálním serverem u společnosti Active24 [19]. Použita byla pouze jediná databáze MS SQL 2008.

Testování započalo v březnu roku 2016. Byla vytvořena testovací otevřená herní instance s neomezenou kapacitou uživatelů. Během pilotního provozu bylo odhaleno a opraveno několik chyb i vyhoveno několika návrhům na zlepšení aplikace, například zobrazení schopností uchazeče o zaměstnání při najetí myši, bez nutnosti zobrazení jeho detailu.

8.3 Propagace

Aplikace byla propagována na sociálních sítích Facebook a Google+, ale pouze nezpлатněnými možnostmi, tedy pouhým sdílením. Mimo to byl umístěn odkaz na komunitní web webtrh.cz, který se převážně zabývá problematikou elektronické komerce a webovými technologiemi.

Společně s aplikací bylo pro propagační účely vytvořeno video, které prezentuje možné operace i grafickou podobu. Video bylo publikováno na serveru [youtube.com](https://www.youtube.com/watch?v=m7KyvbXR_Zo) a lze ho přehrát na adrese https://www.youtube.com/watch?v=m7KyvbXR_Zo. Společně s videem byl publikován i PR článek na serveru vzdelani.cz, který je primárně určen pro studenty i samotné školy. Článek je k dispozici na adrese <http://clanky.vzdelani.cz/vyuka-podnikani-formou-simulacni-hry-tradematch-a5833>.

V rámci práce byl sepsán článek v anglickém jazyce za účelem prezentace vytvořené aplikace na mezinárodní konferenci ICE-B 2016 v Lisabonu, zaměřující se na podnikání i elektronickou komerci. Článek je k dispozici na příloženém cd.

Závěr

Cílem práce bylo navrhnout, implementovat a otestovat simulační hru pro výuku podnikání. Problematika podnikání je značně rozsáhlá a některé její aspekty jsou velice náročné na převod do virtuálního prostředí. V kontextu navrhované aplikace byla problematika značně zúžena a implementována byla pouze funkcionalita, jež je pro uživatele snadno pochopitelná a zábavná s ohledem na obecné principy při vývoji počítačových her. Inspirace byla do jisté míry čerpána u existujících výukových aplikací, ale také u strategických her.

Vývoj aplikace doprovázelo několik problémů, z nichž některé byly v práci zmíněny detailněji. Při jejich řešení bylo přihlédnuto k ověřeným postupům a návrhovým vzorům a aplikace se tak stala snadno rozšiřitelnou.

Výsledné řešení je kombinací výukových a herních prvků, a je zasazené do prostředí podnikání v České republice. Je značně konfigurovatelné a umožňuje tak simulovat různé situace reálného trhu. Podobný přístup se zatím na českém trhu neobjevil, a aplikace má tedy potenciál i k praktickému používání. Oproti ostatním popsaným aplikacím klade vyšší důraz na infrastrukturu firmy jak po grafické stránce, tak po stránce rozhodování. Zaměřuje se na začátky podnikání a nevtahuje uživatele do hry již v roli podnikatele, nýbrž jako prostého člověka, který se podnikatelem teprve stane. Nutí tak uživatele k provádění více dílčích kroků, ať už jde o založení firmy, nebo prvotní vybavování pobočky.

Cílovou skupinou uživatelů jsou převážně studenti středních a vysokých škol se zájmem o hry a podnikání. Aplikace ovšem není vhodná pro lidi, kteří hledají specifické praktické informace o podnikání a není ani vhodná pro lidi, jež hledají hry pro účely odreagování. S postupem času jsou hry stále zjednodušovány. Podle mého názoru se ovšem stále najde dost příznivců her, které jsou tvořeny komplexnějšími pravidly a matematickými principy, jež nutí uživatele přemýšlet trochu více. Dnešní mobilní doba také zapříčiňuje užívání aplikací na zařízeních s malým displejem a s menším výkonem, proto byl kladen velký důraz na implementaci responzivních vlastností.

Použitá literatura

- [1] SRPOVÁ, Jitka. 2010. *Základy podnikání: teoretické poznatky, příklady a zkušenosti českých podnikatelů*. 1. vyd. Praha: Grada, 427 s. ISBN 978-80-247-3339-5.
- [2] *Jak podnikat* [online]. 2016 [cit. 2016-03-31]. Dostupné z: <http://www.jakpodnikat.cz/>
- [3] Základní postup založení společnosti s ručením omezeným. *Business center* [online]. [cit. 2016-03-31]. Dostupné z: <http://business.center.cz/business/pravo/formypodn/sro/zalozeni.aspx>
- [4] *Odvody z mezd ve zkratce - sociální a zdravotní pojištění, daň z příjmu* [online]. [cit. 2016-04-03]. Dostupné z: <http://www.unijob.cz/download/odvody-z-mezd.pdf>
- [5] SCHELL, Jesse. *The art of game design: a book of lenses*. Boca Raton: CRC, c2008. ISBN 978-0-12-369496-6.
- [6] The Strategy Game Designer's Constitution. *Gamedev* [online]. 2011 [cit. 2016-03-31]. Dostupné z: http://www.gamedev.net/page/resources/_/creative/game-design/the-strategy-game-designers-constitution-r2758
- [7] *MARKETHERO2*. 2011. [online]. [cit. 2015-05-13]. Dostupné z: <http://markethero.eu/>
- [8] *MARKETPLACE LIVE*. [online]. [cit. 2015-05-13]. Dostupné z: <http://www.marketplace-simulation.com/>
- [9] *SIMVENTURE*. 2006. [online]. [cit. 2015-05-13]. Dostupné z: <http://simventure.co.uk/>
- [10] *TRAVIAN*. 2004. [online]. [cit. 2016-02-18]. Dostupné z: <http://travian.cz/>
- [11] SEEMANN, Mark. *Dependency injection in .NET*. Shelter Island, NY: Manning, 2012, xxx, 552 p. ISBN 1935182501.
- [12] Creating Isometric Worlds: A Primer for Game Developers. *Tutsplus* [online]. 2013 [cit. 2016-04-21]. Dostupné z: <http://gamedevelopment.tutsplus.com/tutorials/creating-isometric-worlds-a-primer-for-game-developers--gamedev-6511>
- [13] ADAM FREEMAN. *Pro ASP.NET MVC 5 platform*. Berkeley, CA: Apress, 2014. ISBN 9781430265429.
- [14] CEBOLLERO, Miguel, Jay NATARAJAN, Michael COLES a Michael COLES. *Pro T-SQL programmer's guide*. 4th edition. Berkeley, CA: Apress, 2015, xxxiv, 707 pages. ISBN 9781484201466.
- [15] LERMAN, Julia a Rowan MILLER. *Programming Entity Framework: Code First*. 1st ed. Sebastopol, CA: O'Reilly Media, 2012, xi, 176 p. ISBN 1449312942.
- [16] MUNRO, Jamie. *Knockout.js*. 2014. ISBN 978-1-491-91431-1.
- [17] EaselJS api documentation. *CreateJS* [online]. [cit. 2016-02-11]. Dostupné z: <http://www.createjs.com/docs/easeljs/modules/EaselJS.html>
- [18] SPURLOCK, Jake. *Bootstrap*. First edition. Beijing: O'Reilly, 2013, xiv, 109 pages. ISBN 1449343910.

- [19] *Active24 - Domény a profesionální webhosting* [online]. [cit. 2016-03-29]. Dostupné z: <https://www.active24.cz/>

Seznam příloh

Příloha A:	Programátorská příručka	I
------------	-------------------------------	---

Příloha na CD.

Adresářová struktura přiloženého CD:

- Source – Zdrojové kódy aplikace
- Publish – Publikovaná aplikace

Příloha A: *Programátorská příručka*

Postup instalace:

1. Provedení publikace aplikace
2. Vytvoření adresáře s povolenými právy pro zápis a nahrání potřebných souborů
 - a. V publikované aplikaci je tento adresář již vytvořen s názvem „Uploads“ a obsahuje všechny potřebné soubory
3. Konfigurace aplikace (soubor „Web.config“)
 - a. Nastavení připojení k databázi (sekce configuration – connectionStrings):
 - i. connectionString s názvem „DataContext“ zajišťuje přístup k datům výchozí části aplikace
 - ii. connectionString s názvem „DataGameContext“ zajišťuje přístup k datům vzorové aplikace
 - b. Nastavení SMTP (sekce configuration – system.net - mailSettings)
 - c. Nastavení inicializace databáze (sekce configuration – entityFramework – contexts)
 - i. Pro úvodní spuštění je nutné nastavit vlastnost „disableDatabaseInitialization“ u obou kontextů na false, aby byla automaticky vytvořena databázová struktura
 - d. Nastavení vlastností aplikace (sekce configuration – appSettings)
 - i. Hodnota klíče „PageSize“ určuje počet záznamů na jednu stránku a je platná pro celou aplikaci
 - ii. Hodnota klíče „FilesFolder“ určuje cestu k adresáři se soubory (jedná se o adresář zmíněný v kroku 2)
 - iii. Hodnota klíče „AgentKey“ určuje klíč, který musí obsažen jako URL parametr při žádosti o spuštění procedur herní systémové části aplikace

Adresářová struktura publikované aplikace (adresář „Publish“):

- Areas – moduly aplikace
 - Admin – modul administrace
 - Play – modul hry
- bin – binární soubory
- Content – styly a obrázky
- Fonts – písma
- Scripts – javascriptové soubory
- Uploads – soubory potřebné v jednotlivých hrách
- Views – Huml šablony

Adresářová struktura zdrojových souborů (adresář „Source“):

- 3D – zdrojové 3D modely ve formátu blend
- Data, DataGame, Utils, UtilsData, Web – zdrojové soubory jednotlivých komponent

Poznámky k aplikaci:

- V aplikaci je vytvořen administrátorský účet, s přihlašovacími údaji: email: „admin@tradematch.cz“, heslo: „tradematch290416“
- Aplikace je dostupná i na <http://www.tradematch.cz>
- Ladění probíhalo převážně v prohlížeči Google Chrome